SELF-BALANCING ROBOT BERODA DUA DENGAN METODE PID

Agus Setiawan, Deddy Susilo, Gunawan Dewantoro

Program Studi Teknik Elektro, Universitas Kristen Satya Wacana
Salatiga, Indonesia

e-mail: agus.setiawan08899@gmail.com, deddy.susilo@uksw.edu, gunawan.dewantoro@uksw.edu

Abstrak

Perkembangan teknologi membuat para peneliti berkeinginan mengembangkan robot lebih dari dua dekade ini, salah satu pengembangannya yaitu dengan menambahkan fitur *self-balancing* pada robot trainer untuk edukasi. Fitur ini menggunakan konsep pendulum terbalik yang harus menyeimbangkan pusat massa agar dapat berada di posisi seimbang. Untuk dapat membuat robot dapat berdiri dengan seimbang, maka dipakailah sebuah metode yaitu kendali PID. Metode ini bertujuan untuk membuat *error* sudut sekecil-kecilnya sehingga dapat membuat robot beroda dua pada posisi tegak. Robot yang digunakan merupakan robot Trainer Edukasi yang dirancang untuk media pembelajaran dan telah digunakan di beberapa sekolah yang terdapat di Kota Salatiga. Hasil pengujian menunjukkan bahwa robot dapat menyeimbangkan diri dan tahan terhadap gangguan luar dengan baik, dengan *error* kemiringan sebesar 1,14 derajat. Robot ini juga mampu bergerak dalam posisi tegak dengan kecepatan maksimal yang dapat ditangani robot adalah 15,07 cm/detik. Berdasarkan penelitian ini dapat disimpulkan bahwa fitur self-balancing pada robot trainer telah dapat digunakan dengan baik.

Kata kunci: robot keseimbangan, pendulum terbalik, kendali PID, MPU-6050

Abstract

As the technology develops rapidly for the last two decades, researchers seek for developing robots by augmenting self-balancing feature on the educational robot trainer. This feature uses the concept of an inverted pendulum, which is supposed to balance the center of mass such that the robot is in a balanced position. In order for the robot to stay in an upright position, a PID control method is used. This method aims to drive the angle error as small as possible so that it can make a two-wheeled robot in an upright position. The robot used is an Educational Trainer robot designed to be a learning media and has been used in several schools in Salatiga City. The results show that the robot can balance itself and also robust against external disturbances, with an angle error of 1.14 degrees. This robot is also capable of moving in an upright position with a maximum speed of 15.07 cm/second. Based on this research, it can be concluded that the self-balancing feature of the robot trainer can be used properly.

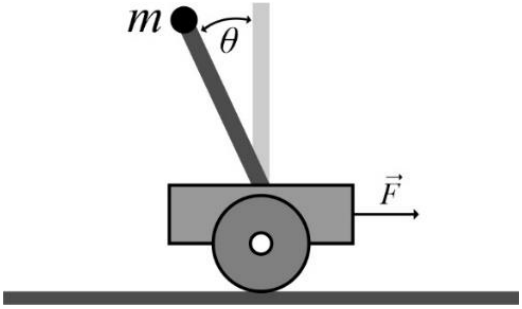
Keywords : *balancing robot, inverted pendulum, PID control, MPU-6050*

PENDAHULUAN

Di dunia modern saat ini, penelitian akan robot semakin berkembang pesat yang disertai dengan teknologi yang terus berkembang juga (Chairunnas and Pamungkas 2018). Hal tersebut membuat pembelajaran di dalam dunia pendidikan juga terus meningkat (Raranda and Rusimamto 2015); (Novandri, Roslidar, and Rahman 2017). Peningkatan tersebut membuat beberapa sekolah di era modern ini menambahkan robotika sebagai materi pembelajaran. Salah satu robot yang digunakan dalam pembelajaran ini adalah Lego Robotic yaitu sebuah produk dari Lego yang dapat dirangkai menjadi sebuah robot dan dapat diprogram sesuai kebutuhan yang diinginkan (Chairunnas and Pamungkas 2018). Dengan perkembangan teknologi ini juga, membuat banyak peneliti berkeinginan untuk mengembangkan robot lebih dari dua dekade ini (Najmurrokhman et al. 2019),

yang salah satunya adalah menambahkan fitur *self-balancing*.

Fitur *self-balancing* ini bekerja sebagai penyeimbang yang di mana konsep awalnya didasari oleh konsep pendulum terbalik (*Inverted Pendulum*) yang harus menyeimbangkan pusat massa, agar dapat berada di posisi seimbang (Chairunnas and Pamungkas 2018); (Raranda and Rusimamto 2015); (Novandri, Roslidar, and Rahman 2017), seperti ditunjukkan Gambar 1. Jika diterapkan ke dalam sistem ini, robot harus menyeimbangkan badan robot agar tidak terjatuh. Untuk menyeimbangkan badan robot tersebut, dibutuhkanlah sebuah metode agar robot dapat berdiri dengan seimbang. Pada penelitian ini, metode pengendalian yang dipilih adalah metode kendali PID (*Proportional, Integral, Derivative*) (Najmurrokhman et al. 2019); (Ma'arif, Puriyanto, and Hasan 2020).



Gambar 1. Pendulum terbalik

Metode kendali PID ini bertujuan untuk membuat keluaran sistem sama dengan setpoint-nya dengan mengolah nilai sinyal kesalahan atau *error* (Shaon et al. 2017); (Kharisma et al. 2018). Di dalam kendali PID terdapat 3 buah kontrol yaitu kendali proporsional (*Proportional Controller*), kendali integral (*Integral Controller*), dan kendali turunan (*Derivative Controller*) yang masing-masing memiliki pengaruh dalam membantu robot agar dapat berdiri tegak (Ma'arif, Puriyanto, and Hasan 2020); (Kharisma et al. 2018). Dalam perancangan sistem kendali PID ini, perlu dilakukan pengaturan parameter Kp, Ki, Kd, di mana pengaturan parameternya akan dilakukan dengan metode *trial and error* agar tanggapan sinyal keluaran sistem terhadap masukan sesuai dengan keinginan. Pada penelitian sebelumnya (Setiawan 2020), telah dilakukan dengan menggunakan metode kendali PID, dan dihasilkan bahwa robot dapat menyeimbangkan diri dengan cukup baik. *Error* kemiringan yang didapat pada robot tersebut yaitu sebesar ±1,89 derajat dan kecepatan robot berjalan dalam keadaan seimbang adalah maksimal 13,47 cm/detik. Dari hasil tersebut, maka pada penelitian ini penulis berkeinginan untuk memodifikasi metode kendali PID yang telah dibuat sehingga didapatkan tingkat keseimbangan yang lebih maksimal atau dapat mencapai 0 derajat yang akan diterapkan pada robot Trainer Edukasi.

Robot Trainer Edukasi ini merupakan sebuah robot yang akan digunakan sebagai media pengaplikasian fitur *self-balancing* dengan metode PID (Setiawan 2020), seperti ditunjukkan oleh Gambar 2. Robot ini dirancang dengan tujuan sebagai media pembelajaran atau pendidikan yang dapat digunakan untuk berbagai jenjang pendidikan. Target dari penggunaan robot ini yaitu dapat digunakan dari jenjang Sekolah Dasar, Sekolah Menegah Pertama, dan Sekolah Menegah Atas. Saat ini, robot Trainer Edukasi telah digunakan di beberapa sekolah yang ada di Kota Salatiga provinsi Jawa Tengah, beberapa sekolah yang telah menggunakan robot ini yaitu SMP Laboratorium Kristen Satya Wacana Salatiga, SMP Muhammadiyah Plus Salatiga, SMP Anak Terang Salatiga dan SMP Kristen 2 Salatiga.

Gambar 2. Robot trainer edukasi

Pada pembuatan robot ini, komponen utama yang akan digunakan yaitu MPU6050 atau IMU (*Inertial Measurement Unit*) Sensor, di mana terdapat dua buah sensor yaitu sensor *accelerometer* yang digunakan untuk mendeteksi kemiringan serta sensor *gyroscope* yang digunakan untuk mendeteksi kecepatan sudut apabila robot akan terjatuh (Shaon et al. 2017), (Zaini, Khoswanto, and Pasila 2017). Sementara itu, mikrokontroler akan menggunakan Arduino Nano sebagai komponen untuk memproses sinyal-sinyal yang didapat dari IMU sensor untuk merealisasikan kendali PID pada robot.

METODE

Complementary Filter

Complementary Filter adalah salah satu metode yang digunakan untuk melakukan *filtering*. Fungsi dari filter ini yaitu untuk meminimalisir nilai yang dikeluarkan oleh sensor agar memiliki *noise* yang kecil sehingga dapat menghasilkan nilai keluaran yang akurat. Filter ini merupakan gabungan dari *high-pass filter* yang berasal dari keluaran sensor *gyroscope* MPU6050 dan *low-pass filter* yang berasal dari keluaran sensor *accelerometer* MPU6050 (Fahmi, Maulana, and Kurniawan 2017), untuk rumus *complementary filter* ditunjukkan pada Persamaan (1). Hasil dari gabungan filter ini berupa nilai sudut yang selanjutnya akan digunakan untuk melakukan perhitungan kendali PID.

$$Sudut = (\alpha)*(Sudut+out\_gyro*dt)+(1-\alpha)*(out\_acc) \quad (1)$$

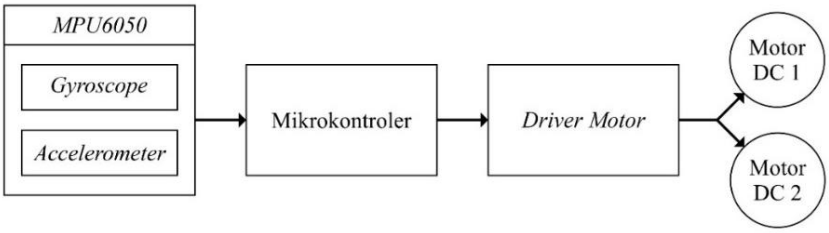
α = koefisien filter
out_gyro = keluaran gyroscope
dt = delta time
out_acc = keluaran accelerometer

Kendali PID

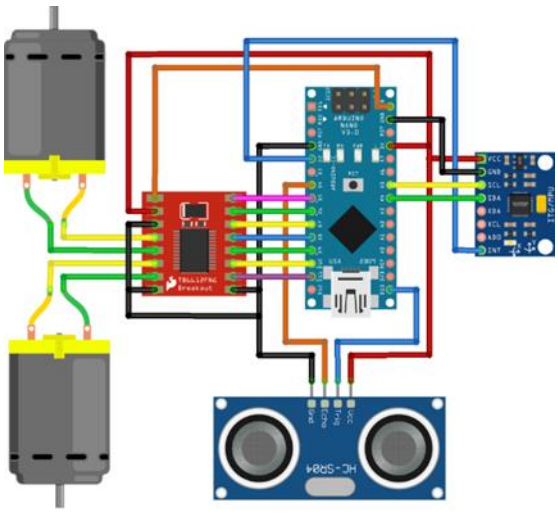
Kendali PID (*Propotional, Integral, Derivative*) adalah salah satu metode kontrol klasik yang sering digunakan oleh industri karena memiliki struktur yang sederhana dan stabil (Solihin and Endryansyah 2020). Sistem kendali ini melakukan kontrol dengan menggunakan 3 komponen yaitu P, I, dan D. Ketiga komponen ini memiliki tanggung jawab masing-masing, di mana komponen P akan bertanggung jawab untuk nilai kesalahan saat ini, komponen I akan bertanggung jawab untuk nilai kesalahan sebelumnya, dan komponen D akan bertanggung jawab untuk kemungkinan kesalahan mendatang (Novandri, Roslidar, and Rahman 2017). Kendali PID ini hanya membutuhkan data masukan dari proses yang diukur, oleh karena itu perlu dilakukan penalaan (*tuning*) terhadap parameter untuk menghasilkan respon sistem yang diinginkan

Perancangan Sistem

Perancangan sistem pada *self-balancing* robot ini akan dimulai dengan melakukan perancangan *hardware* robot. Untuk mempermudah perancangan *hardware*, maka digunakan diagram blok seperti yang ditunjukkan Gambar 3.



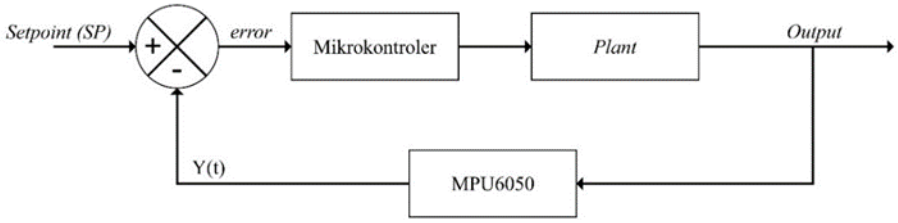
Gambar 3. Diagram blok robot



Gambar 4. Skematik *wiring* Arduino

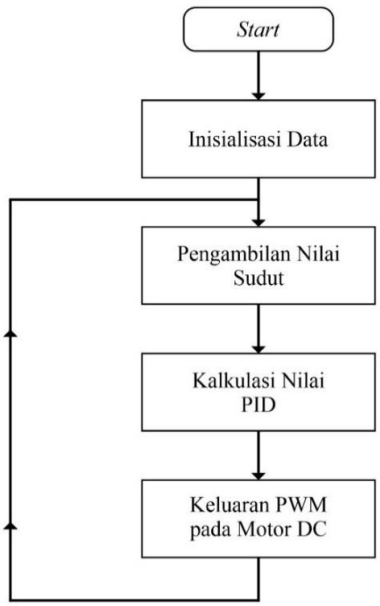
Dari diagram blok tersebut, selanjutnya akan dibuat rancangan skematik *wiring* Arduino yang berguna untuk mempermudah proses perancangan robot, seperti ditunjukkan Gambar 4. Di dalam rancangan skematik *wiring* Arduino, terdapat penambahan komponen yaitu sensor ultrasonik. Sensor ultrasonik ini akan digunakan sebagai *switch* untuk membantu proses pengujian kemampuan robot berjalan dalam keadaan seimbang.

Setelah selesai melakukan perancangan *hardware*, maka langkah selanjutnya adalah melakukan perancangan *software* robot. Pada perancangan software ini, lebih ditujukan untuk perancangan program robot. Oleh karena itu, langkah pertama yang akan dilakukan adalah merancang sistem kendali *closed-loop* dan diagram alir robot guna untuk mempermudah proses pembuatan program pada robot, seperti ditunjukkan Gambar 5 dan 6.



Gambar 5. Sistem kendali *closed-loop*

Setpoint (*SP*) = nilai awal sebagai pembanding.
Y(*t*) = nilai keluaran dari MPU6050 setiap waktu.
error = nilai kesalahan sudut yang didapat dari *Setpoint* – Y(*t*)



Gambar 6. Diagram alir robot

Dengan memiliki diagram kendali dan diagram alir robot, maka dapat dibuatlah program untuk robot *self-balancing* ini. Dengan nilai sudut (Y(*t*)) sebagai dasar yang akan digunakan pada perhitungan nilai kendali PID, dan selanjutnya hasil dari perhitungan kendali PID akan menjadi *output* nilai pwm pada motor DC.

HASIL DAN PEMBAHASAN

Penalaan Konstanta PID

Agar robot dapat berdiri dengan seimbang, maka perlu dilakukan penalaan untuk mendapatkan parameter terbaik. Tabel 1 – 3 menunjukkan nilai-nilai konstanta PID yang didapat beserta respon yang terjadi ke robot.

Tabel 1. Hasil penalaan konstanta Kp

Percobaan	Kp	Ki	Kd	Respon
1	1	0	0	Robot terjatuh karena motor tidak bergerak
2	10	0	0	Robot terjatuh karena kecepatan gerak motor sangat lambat
3	20	0	0	Robot mulai bisa seimbang tetapi masih terjatuh karena kecepatan gerak motor masih terlalu lambat
4	25	0	0	Robot seimbang tetapi terkadang masih terjatuh karena kecepatan gerak motor yang masih lambat
5	30	0	0	Robot seimbang dengan lebih sedikit kemungkinan untuk terjatuhnya
6	35	0	0	Robot seimbang tetapi mulai terjatuh lagi karena kecepatan gerak motor sedikit lebih cepat
7	40	0	0	Robot mampu seimbang tetapi mulai terjatuh lagi karena kecepatan gerak motor semakin cepat
8	50	0	0	Robot mampu seimbang tetapi hanya sesaat karena kecepatan gerak motor terlalu cepat
9	60	0	0	Robot mampu seimbang tetapi hanya sesaat karena kecepatan gerak motorterlalu cepat
10	70	0	0	Robot mampu seimbang tetapi hanya sesaat karena kecepatan gerak motor sangat cepat

Tabel 2. Hasil penalaan konstanta Ki

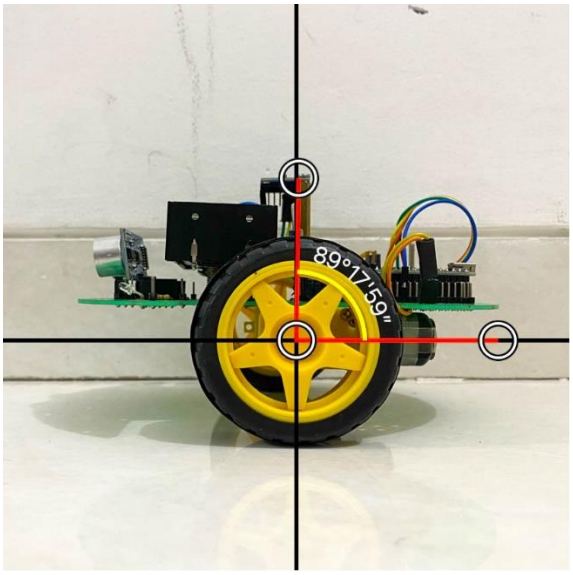
Percobaan	Kp	Ki	Kd	Respon
1	30	10	0	Kemungkinan untuk jatuh besar karena respon robot sangat lambat
2	30	20	0	Kemungkinan untuk jatuh besar karena respon robot masih sangat lambat
3	30	30	0	Kemungkinan untuk jatuh besar karena respon robot masih sangat lambat
4	30	40	0	Kemungkinan untuk jatuh semakin kecil karena respon robot semakin cepat
5	30	50	0	Kemungkinan untuk jatuh tidak ada tetapi respon robot masih terlalu lambat
6	30	60	0	Kemungkinan untuk jatuh tidak ada tetapi respon robot masih terlalu lambat
7	30	70	0	Kemungkinan untuk jatuh tidak ada tetapi respon robot masih terlalu lambat
8	30	75	0	Kemungkinan untuk jatuh tidak ada
9	30	80	0	Kemungkinan untuk jatuh bertambah karena respon robot semakin cepat
10	30	90	0	Kemungkinan untuk jatuh besar karena respon robot sangat cepat

Tabel 3. Hasil penalaan konstanta Kd

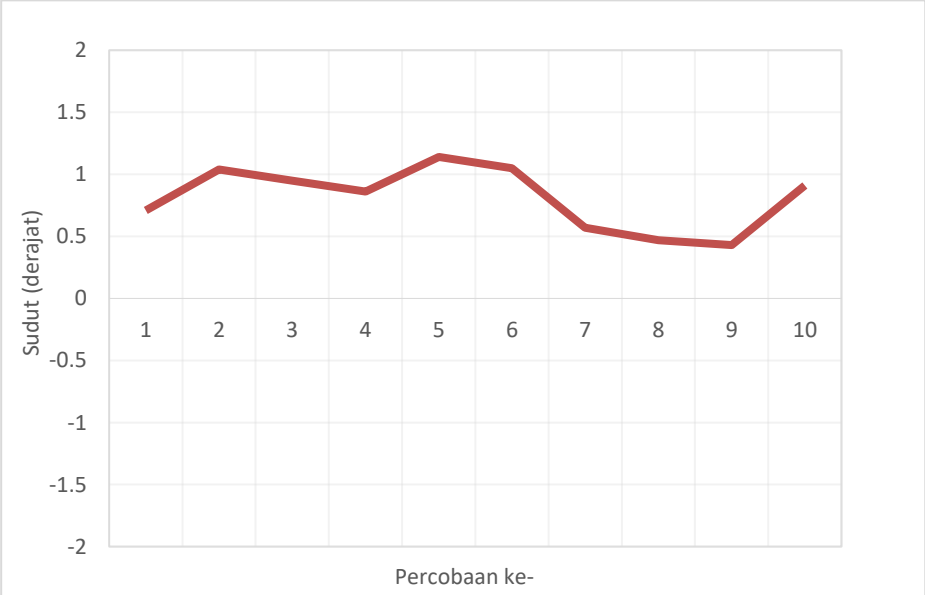
Percobaan	Kp	Ki	Kd	Respon
1	30	75	0,1	Perputaran roda terlalu lambat sehingga robot susah untuk mendapatkan posisi seimbang
2	30	75	0,3	Perputaran roda terlalu lambat sehingga robot susah untuk mendapatkan posisi seimbang
3	30	75	0,4	Perputaran roda masih terlalu lambat sehingga robot masih susah untuk mendapatkan posisi seimbang
4	30	75	0,5	Perputaran roda tepat tetapi robot masih sedikit susah untuk mendapatkan posisi seimbang
5	30	75	0,6	Perputaran roda tepat sehingga robot mudah mendapatkan posisi seimbang
6	30	75	0,7	Perputaran roda tepat tetapi robot masih sedikit susah untuk mendapatkan posisi seimbang
7	30	75	0,8	Perputaran roda semakin cepat sehingga robot semakin susah untuk mendapatkan posisi seimbang
8	30	75	1	Perputaran roda terlalu cepat sehingga robot sudah mendapatkan posisi seimbang serta sangat bergetar
9	30	75	2	Perputaran roda terlalu cepat sehingga robot sudah mendapatkan posisi seimbang serta sangat bergetar
10	30	75	5	Perputaran roda terlalu cepat sehingga robot sudah mendapatkan posisi seimbang serta sangat bergetar

Setelah melakukan penalaan parameter PID, maka didapatlah nilai-nilai konstanta PID yang akan membuat robot dapat berdiri dalam keadaan seimbang. Konstanta PID yang didapat tersebut adalah Kp=30, Ki=75, dan Kp=0,6. Selanjutnya dengan menggunakan nilai konstanta PID tersebut akan dilakukan pengukuran *error* kemiringan pada robot. Proses pengukuran *error* kemiringan robot ditunjukkan seperti pada Gambar 7. Pengukuran tersebut dilakukan menggunakan aplikasi *angle meter* yang hasilnya ditampilkan dalam bentuk derajat, menit, detik. Hasil tersebut selanjutnya dikonversi ke dalam bentuk desimal dengan satuan derajat agar dapat

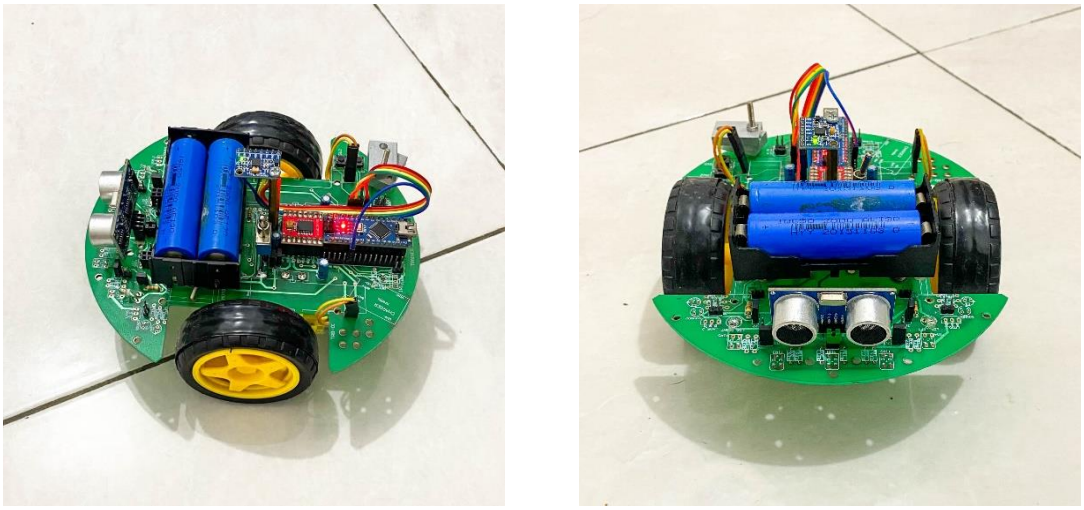
dihitung nilai selisih *error* kemiringan pada robot. Hasil perhitungannya ditunjukkan pada Gambar 8. Dari pengukuran tersebut, maka didapatlah *error* kemiringan maksimum pada robot yaitu sebesar 1,14 derajat. Hasil realisasi robot ditunjukkan oleh Gambar 9.



Gambar 7. Proses pengukuruan *error* kemiringan robot



Gambar 8. Grafik pengukuran *error* kemiringan robot



(a) (b)

Gambar 9. Realisasi robot (a) Tampak samping (b) Tampak depan

Pengujian Keseimbangan Robot

Pengujian keseimbangan robot akan dilakukan dengan menggunakan konstanta PID yang telah didapatkan, serta akan diberikan gangguan dari luar berupa dorongan seperti menekan posisi depan atau belakang robot yang selanjutnya akan dicatat keseimbangannya untuk setiap percobaan yang dilakukan. Tabel 4 menunjukkan hasil pengujian keseimbangan robot yang telah dilakukan.

Tabel 4. Pengujian keseimbangan robot dengan diberi gangguan

Percobaan	Keterangan
1	Robot Seimbang
2	Robot Seimbang
3	Robot Seimbang
4	Robot Seimbang
5	Robot Seimbang
6	Robot Seimbang
7	Robot Seimbang
8	Robot Seimbang
9	Robot Seimbang
10	Robot Seimbang

Setelah dilakukan pengujian, maka hasilnya adalah dengan menggunakan konstanta PID yang telah didapatkan, robot mampu untuk menyeimbangkan diri kembali ke keadaan seimbang saat diberikan gangguan dari luar maupun tidak ada gangguan dari luar dengan persentase keberhasilan 100%.

Pengujian Kecepatan Robot Berjalan dalam Keadaan Seimbang

Pengujian PWM untuk robot dilakukan sebanyak 3 kali dengan menggunakan nilai PWM yang berbeda-beda hingga didapatkan nilai PWM yang sesuai dengan target yang telah ditentukan. Jarak yang digunakan adalah 40 cm yang menjadi acuan untuk mengukur waktu tempuh robot. Gambar 10 menunjukkan hasil percobaan saat robot berjalan.

Gambar 10. Grafik percobaan robot berjalan

Setelah dilakukan pengujian, didapat nilai PWM yang digunakan yaitu 120, 130 dan 140, di mana PWM = 140 merupakan nilai PWM tertinggi supaya robot tidak jatuh. Setelah mengetahui nilai PWM yang digunakan dan hasil kecepatan untuk setiap PWM, maka selanjutnya akan dilakukan perhitungan waktu rata-rata robot menempuh jarak 40 cm dari 10 kali percobaan yang telah dilakukan. Dimulai dari PWM = 120, didapatkan waktu rata-ratanya adalah 2,97 detik, untuk PWM = 130, didapatkan waktu rata-ratanya adalah 2,803 detik, dan untuk PWM = 140, didapatkan waktu rata-ratanya adalah 2,655 detik. Setelah didapat waktu rata-ratanya, maka dapat disimpulkan kecepatan rata-rata robot untuk setiap PWM. Kecepatan rata-rata robot dengan PWM = 120, 130, dan 140 berturut-turut adalah 13,47 cm/detik, 14,27 cm/detik, dan 15,07 cm/detik.

SIMPULAN

Dalam pembuatan *Self-Balancing* Robot beroda dua dengan metode PID dapat disimpulkan, bahwa dengan menggunakan nilai konstanta PID yaitu Kp=30, Ki=75, dan Kd=0,6 berhasil membuat robot dapat berdiri dengan tegak dan seimbang, dengan *error* kemiringan yang didapat sebesar 1,14 derajat. Serta telah dilakukan percobaan dengan memberikan gangguan dari luar berupa dorongan, seperti menekan posisi depan atau belakang robot, dan robot tetap dapat menyeimbangkan diri kembali. Pada penelitian ini juga ditambahkan fitur robot dapat berjalan dalam keadaan seimbang. Setelah dilakukan pengujian, menghasilkan kecepatan maksimum yang dapat ditangani oleh robot adalah 15,07 cm/detik.

DAFTAR PUSTAKA

Chairunnas, Andi, and Triyoga Ginanjar Pamungkas. 2018. “Sistem Kontrol Robot Penyeimbang Berbasis Arduino Menggunakan Metode PID Dengan Komunikasi Bluetooth HC-05.” 15(02): 140–51.

Raranda, and Puput Wanarti Rusimamto. 2015. “Implementasi Kontroler PID Pada Two Wheels Self Balancing Robot Berbasis Arduino Uno.” 06(02): 89–96.

Novandri, Andri, Roslidar, and Aulia Rahman. 2017. “Rancang Bangun Robot Self Balancing Berbasis Mikrokontroler Atmega328P Dengan Kendali PID.” 02(02): 15–23.

Najmurrokhman, Asep, Kusnandar, Irfan Irfansyah, and Ahmad Daelami. 2019. “Rancang Bangun Auto Balancing Robot Menggunakan Metode Kendali PID.” 05(02): 15–23.

Ma'arif, Alfian, Riky Dwi Puriyanto, and Fadlur T Hasan. 2020. "Robot Keseimbangan Proporsional-Integral-Derivatif (PID) Dan Kalman Filter." 04(02): 117–27.

Shaon, A K M Ashfaque Shakil et al. 2017. "Design and Implementaion of a Self-Balancing Robot."

Kharisma, Oktaf Brillian, Ahmad Wildan, Auliaullah, and Folkes E Laumal. 2018. "Implementasi Sensor MPU6050 Untuk Mengukur Kesetimbangan Self Balancing Robot Menggunakan Kontrol PID." : 357–64.

Setiawan, Agus. 2020. "Perancangan Self Balancing Robot Beroda Dua Dengan Metode PID."

Zaini, Indro, Handry Khoswanto, and Felix Pasila. 2017. "Prototipe Balancing Robot Dengan Metode Kendali PID." 10(01): 34–39.

Fahmi, Hafizhuddin Zul, Rizal Maulana, and Wijaya Kurniawan. 2017. "Implementasi Complementary Filter Menggunakan Sensor Accelerometer Dan Gyroscope Pada Keseimbangan Gerak Robot Humanoid." 01(11): 1376–84.

Solihin, Andrian Kusuma, and Endryansyah. 2020. "Rancang Bangun Sistem Kontrol PID Untuk Pengendalian Kecepatan Prototipe Lift Berbasis Labview." 09(01): 893–901.