ISSN 1979-9780

JURNAL elektro

JURNAL ELEKTRO	Vol. 9	No. 2	Hlm. 69-140	Okt 2016	ISSN 1979-9780

Terbit dua kali setahun

TWO WHEELS SELF-BALANCING ROBOT BERBASIS *ARDUINO NANO* MENGGUNAKAN METODE PID

Mahardi Arifin[1], Budi Harsono[2]
[1,2]Program Studi Teknik Elektro Fakultas Teknik dan Ilmu Komputer
Universitas Kristen Krida Wacana
e-mail: [1]mahardi.arifin@yahoo.com, [2]budi.harsono@ukrida.ac.id

ABSTRAK

Self-balancing robot merupakan robot yang dapat menjaga keseimbangan dirinya sendiri. *Two wheels self-balancing robot* memiliki roda di kedua sisinya dan dapat menyeimbangkan dirinya dengan mengatur kecepatan motor berdasarkan sensor *gyroscope* dan *accelerometer*. Terdapat beberapa teknik kontrol yang dapat digunakan untuk *self-balancing robot*, yaitu *proportional-integral-derivative* (PID), *fuzzy logic*, dan *linear quadratic regulator* (LQR). Dalam penelitian ini dibuat *two wheels self-balancing robot* dengan mikrokontroler *Arduino Nano* menggunakan metode kontrol PID. Sensor yang digunakan adalah MPU-6050 yang merupakan gabungan *gyroscope* dan *accelerometer*. *Complementary filter* digunakan untuk menggabungkan data kedua sensor tersebut dan untuk mengontrol motor DC yang ada pada robot digunakan *driver* motor L293D.

Kata kunci: robot, *self-balancing*, *Arduino*, MPU-6050, PID

ABSTRACT

Self-balancing robot is a robot that can maintain its balance. Two wheels self-balancing robot have wheels on each side and can balance itself by controlling motor's velocity based on gyroscope and accelerometer sensor. There are some control method for self-balancing robot such as proportional-integral-derivative (PID), fuzzy logic, and linear quadratic regulator (LQR). In this research, two wheels self-balancing robot based on Arduino Nano using PID control is built. MPU-6050, which is combination of gyroscope and accelerometer sensor, is used as the sensor. Complementary filter is used to merge the data from two sensors and to control the DC motors in the robot, L293D motor driver is used.

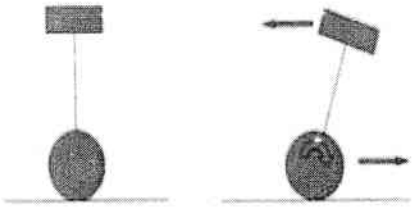
Keywords: robot, *self-balancing*, *Arduino*, MPU-6050, PID

PENDAHULUAN

Permasalahan pendulum terbalik bukan merupakan hal yang asing di dalam bidang teknik kontrol. Keunikan dan aplikasi yang luas dari sistem yang tidak stabil telah menarik perhatian peneliti dan penggemar robotik di seluruh dunia. Beberapa tahun ini, para peneliti telah mengaplikasikan ide *mobile inverted pendulum* untuk berbagai permasalahan seperti merancang alur gerak robot *humanoid*, kursi roda robotik, dan sistem transportasi personal [5].

Dasar untuk membuat robot beroda dua dapat setimbang adalah dengan cara mengendalikan roda searah dengan arah jatuhnya bagian atas sebuah robot. Apabila proses tersebut dapat terlaksana maka robot tersebut dapat setimbang.

Saat *balancing robot* beroda dua condong ke depan atau miring ke kanan seperti Gambar 1, maka tindakan yang perlu dilaksanakan adalah motor bergerak searah dengan arah kemiringan yang terjadi, sehingga robot akan kembali tegak lurus dengan permukaan bidang datar. Gaya yang digunakan untuk menyeimbangkan robot didapat dari putaran roda yang dihasilkan dari motor [4].

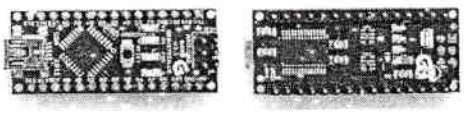


Gambar 1 *Balancing robot* beroda dua menyeimbangkan diri

LANDASAN TEORI

A. *Arduino Nano*

Arduino Nano adalah *board* kecil, lengkap, dan *breadboard-friendly* berbasis ATmega328 (*Arduino Nano* 3.x) atau Atmega168 (*Arduino Nano* 2.x). *Arduino Nano* tidak memiliki *power jack* DC, dan bekerja menggunakan Mini-B USB. Nano didesain dan diproduksi oleh Gravitech.



Gambar 2 *Arduino nano*

Arduino Nano memiliki beberapa fasilitas untuk berkomunikasi dengan komputer, *Arduino* lain, maupun mikrokontroler lain. ATmega168 dan Atmega328 dapat melakukan komunikasi serial UART TTL (5V) yang tersedia pada digital pin 0 (Rx) dan 1 (Tx). FTDI FT232RL pada *board* menjadi saluran komunikasi serial menggunakan USB dan FTDI *driver* (tergabung dengan *software Arduino*) untuk berkomunikasi dengan komputer memanfaatkan *virtual com port*. Spesifikasi dari *Arduino Nano* ditunjukkan pada Tabel 1 [1].

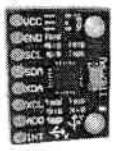
Tabel 1 Spesifikasi *arduino nano*

Mikrokontroler	Atmel ATmega168 atau ATmega328
Tegangan kerja (*logic level*)	5 V
Tegangan masuk (*recommended*)	7-12 V
Tegangan masuk (*limits*)	6-20 V
Pin I/O Digital	14 (dengan 6 PWM *output*)
Pin *Input* Analog	8
Arus DC per I/O Pin	40 mA
Flash Memory	16 kB (ATmega168) atau 32 kB (ATmega328) dengan 2 kB terpakai untuk *bootloader*
SRAM	1 kB (ATmega168) atau 2 kB (ATmega328)
EEPROM	512 *bytes* (ATmega168) atau 1 kB (ATmega328)
Clock Speed	16 MHz
Dimensi	0.73" x 1.70"
Panjang	45 mm
Lebar	18 mm
Berat	5 g

B. MPU-6050

MPU-6050 merupakan sensor dari *InvenSense* yang berisi MEMS *accelerometer* dan MEMS *gyroscope* dalam satu *chip*. MPU-6050 memiliki tingkat akurasi yang tinggi, dengan 16

bit analog to digital converter pada setiap *channel* serta *channel* untuk x, y, dan z yang dapat ditangkap bersamaan. Sensor menggunakan komunikasi I^2C untuk berkomunikasi dengan *Arduino*. Terdapat juga kombinasi MPU-6050 dengan magnetometer (kompas) dalam satu *chip*, yaitu MPU-9150 [3].



Gambar 3 MPU-6050

MPU-6050 memiliki tiga MEMS *rate gyroscope* independen, yang mendeteksi perputaran dalam sumbu x, y, dan z. Saat *gyro* berputar dalam sumbu tersebut, *Coriolis effect* menyebabkan getaran yang dideteksi oleh *capacitive pickoff*. Sinyal keluaran kemudian diamplifikasi, didemodulasi, dan difilter untuk menghasilkan tegangan yang proporsional dengan laju sudut. Tegangan ini kemudian diubah menjadi data digital melalui *on-chip 16-bit Analog-to-Digital Conveter* (ADC) untuk masing-masing sumbunya. *Full scale range* pada sensor *gyro* dapat diprogram secara digital pada nilai ±250, ±500, ±1000, or ±2000 *degrees per second* (dps). *Sample rate* ADC pada *gyroscope* dapat diprogram dari 8000 *samples per second* sampai dengan 3.9 *samples per second*.

Three-axis accelerometer pada MPU-6050 menggunakan *proof mass* terpisah untuk setiap sumbunya. Percepatan pada sumbu tertentu menimbulkan perpindahan pada *proof mass* yang bersesuaian dan sensor *capacitive* mendeteksi perpindahan ini secara berbeda. Arsitektur pada MPU-6050 mengurangi kerentanan *accelerometer* terhadap *fabrication variations* dan juga *thermal drift*. Saat *device* diletakkan pada permukaan rata, maka akan terukur 0 *g* pada sumbu x dan y, serta 1 *g* pada sumbu z. Skala pada *accelerometer* telah dikalibrasi oleh pabrik dan nilainya independen dari tegangan suplai. Setiap sensor memiliki ADC sigma-delta untuk menghasilkan *output* digital. *Full scale range* dari *output* dapat diatur pada ±2*g*, ±4*g*, ±8*g*, or ±16*g*.

C. *Complementary Filter*

Penggunaan *accelerometer* dan *gyroscope* memiliki tujuan yang sama, yaitu untuk mendapatkan posisi sudut dari objek. *Gyroscope* dapat melakukan ini dengan cara mengintegral kecepatan sudut terhadap waktu. Posisi sudut dari *accelerometer* didapatkan dengan menentukan posisi gaya gravitasi (*g-force*) yang selalu terbaca pada *accelerometer*. Besaran sudut didapatkan dengan menggunakan fungsi *atan2* yang dapat dirumuskan sebagai Persamaan (1) [8].

$$Acc = atan2(AcX, \sqrt{(AcY^2+AcZ^2)}) \quad (1)$$

dengan

AcX : *data accelerometer* sumbu x.

AcY : *data accelerometer* sumbu y.

AcZ : *data accelerometer* sumbu z.

Walaupun hasil kedua sensor tersebut sudah dalam besaran sudut, tetapi pada masing-masing sensor masih terdapat masalah yang membuat data tidak dapat digunakan tanpa adanya filter.

Accelerometer mengukur seluruh gaya yang bekerja pada benda, termasuk gaya di luar gaya gravitasi. Setiap gaya kecil yang bekerja pada benda dapat mengganggu pengukuran sepenuhnya. Jika ada gerakan dari benda, gaya gerak

dari benda tersebut juga akan terukur oleh sensor. Karena itu, data dari *accelerometer* hanya dipergunakan dalam jangka panjang.

Berbeda dengan *accelerometer*, data dari *gyroscope* tidak terpengaruh oleh gaya luar. Tetapi *gyroscope* sendiri memliki kekurangan. Dengan adanya integral terhadap waktu, pengukuran cenderung mengalami *drift*, yaitu nilai tidak kembali pada nol saat sistem dikembalikan pada posisi awal. Data dari *gyroscope* hanya dapat digunakan pada jangka pendek, dan akan semakin terkena *drift* dalam jangka panjang.

Complementary filter dapat menjadi salah satu solusi sederhana untuk menjembatani kedua permasalahan tersebut. Dalam jangka pendek, digunakan data *gyroscope*, karena data yang presisi dan tidak terganggu gaya luar. Dalam jangka panjang, digunakan data *accelerometer*, sehingga tidak data tidak mengalami *drift*. Secara sederhana, *complementary filter* untuk mendapatkan data sudut dapat dirumuskan seperti Persamaan (2) [6].

$$Angle = 0{,}98\times(Angle+Gyro\times dt)+0{,}01\times Acc \quad (2)$$

dengan:

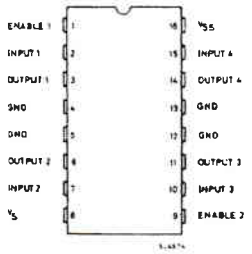
Angle:data sudut.

Gyro : data *gyroscope* setelah dibagi dengan nilai sensitivitas *gyroscope*.

Acc : data *accelerometer* setelah diproses dengan fungsi *atan2*.

D. L293D

L293D merupakan IC yang berfungsi sebagai *driver* motor DC untuk menyalakan dan mematikan motor dalam satu arah dan juga dapat digunakan untuk mengendalikan dalam dua arah. Untuk memudahkan pemakaian sebagai *bridge*, setiap *channel* dilengkapi dengan *enable input*. Sumber *input* terpisah untuk *logic*, membuat pengoperasian dilakukan pada tegangan yang lebih rendah. L293D memiliki 16 *pin* dengan empat *pin* tengah terkoneksi bersama sebagai *heatsink* dan *ground*.



Gambar 4 Koneksi *pin* pada L293D

Keterangan:

Enable 1,2	:	Dikoneksi dengan Arduino untuk menyalakan dan mematikan motor
Input 1,2,3,4	:	Dikoneksi dengan Arduino untuk mengontrol arah motor
Output 1,2,3,4	:	Dikoneksi dengan motor DC
GND	:	*Ground*
Vs	:	Sumber tegangan motor
Vss	:	Sumber tegangan IC

Untuk dapat mengendalikan motor dalam dua arah (*bidirectional*), *output* 1 dan 2 disambungkan pada satu motor, sehingga membentuk *full H-bridge* yang arahnya diatur oleh *input* 1 dan 2. Hal yang sama juga digunakan untuk *output* 3 dan 4 [7].

Tabel 2 Tabel nilai *logic* kontrol motor secara *bidirectional*

EN	IN1	IN2	Fungsi
H	H	L	Motor berputar ke kiri
H	L	H	Motor berputar ke kanan
H	H	H	*Fast motor stop*
H	L	L	*Fast motor stop*
L	X	X	*Free-running motor stop*

E. Kontrol PID

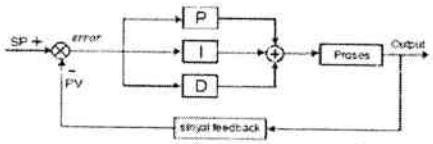
Pengendali PID adalah suatu sistem pengendali yang merupakan gabungan antara pengendali proporsional, integral, dan turunan (*derivative*). Karakteristik PID *controller* sangat dipengaruhi oleh kontribusi besar dari ketiga parameter, yaitu P, I, dan D. Proporsional *controller* (Kp) akan memberikan efek mengurangi waktu naik, tetapi tidak menghapus kesalahan keadaan tunak, Integral *controller* (Ki) akan memberikan efek menghapus keadaan tunak, tetapi berakibat memburuknya respons transien, diferensial *controller* (Kd) akan memberikan efek meningkatnya stabilitas sistem, mengurangi *over-shoot*, dan menaikan respons transfer.

Dalam pengaplikasian, masing-masing pengendali dapat berdiri sendiri atau dapat melakukan pengombinasian. Dalam perancangan sistem kontrol PID yang perlu dilakukan adalah mengatur parameter Kp, Ki atau Kd agar tanggapan sinyal keluaran sistem sesuai dengan yang diinginkan. Dalam waktu kontinyu, sinyal keluaran pengendali PID dirumuskan pada Persamaan (3).

$$u(t) = Kp \times e(t) + Ki \times \int e(t)\, dt + Kd \frac{de(t)}{dt} \quad (3)$$

dengan:

u(t) : sinyal keluaran pengendali PID
Kp : konstanta proporsional
Ki : konstanta integral
Kd : konstanta turunan
e(t) : sinyal kesalahan = referensi – *output*

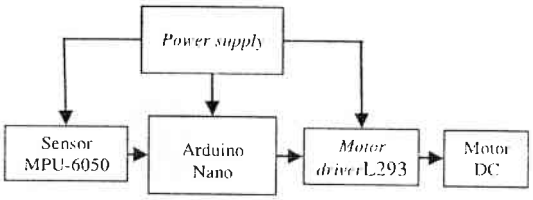


Gambar 5 Diagram blok sistem kontrol PID

SP adalah nilai *set point*, nilai referensi yang diinginkan. *Error* dihasilkan ketika nilai *output* sistem berbeda dengan nilai *set point*. Sinyal *error* ini kemudian diproporsional, diintegral, dan atau didiferensial; hingga sinyal *output* bernilai sama dengan nilai *set point* [2].

METODE PENELITIAN

Diagram blok *two wheels self-balancing robot* ditunjukkan pada Gambar 6.

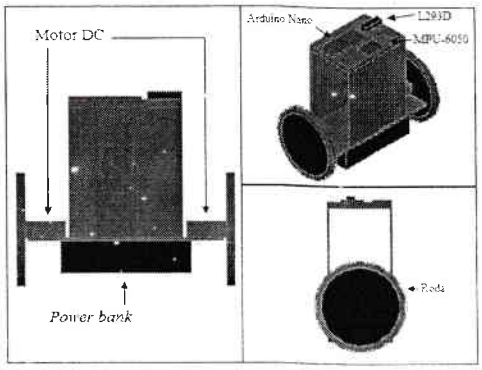


Gambar 6 Diagram blok *two wheels self-balancing robot*

Robot terdiri dari badan robot serta roda di kanan dan kirinya. *Arduino Nano* dipilih sebagai kontroler karena ukurannya yang kecil, harganya yang murah, dan jumlah *port input/output* yang cukup untuk digunakan. Sensor yang digunakan adalah MPU-6050 yang merupakan gabungan sensor *gyroscope* dan *accelerometer*. MPU-6050 memiliki *gyroscope* dan *accelerometer* tiga axis dan sensitivitas yang cukup untuk digunakan pada *two wheels self-balancing robot*. *Complementary filter* digunakan untuk mendapatkan data gabungan dari dua sensor tersebut untuk dipakai sebagai acuan kontrol robot. *Complementary filter* merupakan metode penggabungan data sensor *gyroscope* dan *accelerometer* yang sederhana, tetapi cukup untuk digunakan pada *two wheels self-balancing robot*. Robot menggunakan IC L293D sebagai *driver* motor DC. *Power bank* digunakan sebagai sumber tegangan 5V

untuk robot dan motor. Untuk menjaga kestabilan tegangan, dipasang kapasitor pada motor.

Diagram alir yang digunakan untuk program kontrol robot ditunjukkan pada Gambar 8, dengan nilai Kp=3, Kd=15, dan Ki=10 merupakan nilai *tuning* PID awal.



Gambar 7 Rancangan *hardware two wheels self-balancing robot*

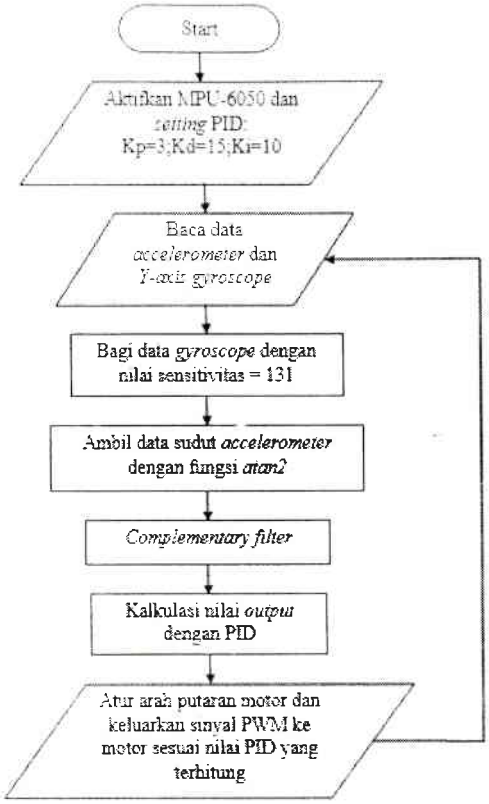
ANALISIS DAN PEMBAHASAN

A. Pengujian Sensor

Hasil pengujian sensor ditujukan pada Tabel 3.

Pengujian sensor dilakukan dengan membandingkan hasil pengukuran sudut kemiringan robot secara manual dengan hasil *angle*, yaitu sudut hasil pembacaan *gyroscope* dan *accelerometer* yang telah diproses melalui *complementary filter*. Sudut diukur terhadap sumbu z positif dalam satuan derajat.

Dari hasil pengukuran didapatkan bahwa pembacaan data sensor yang digunakan untuk perhitungan sudah mendekati sudut aslinya. Pembacaan sensor memiliki batas pembacaan sampai mendekati 90° (dan -90° untuk nilai sudut negatif) diakibatkan penggunaan rumus *atan2* pada *accelerometer* selalu mempunyai nilai positif pada penyebutnya.



Gambar 8 Diagram alir program *two wheels self-balancing robot*

Tabel 3 Hasil pengujian sensor

Sudut terukur (°)	*Angle* (°)
0	0,32
10	10,44
20	20,15
30	30,64
40	40,64
50	50,57
60	60,16
70	70,17
80	80,56
90	84,73
-10	-10,07
-20	-20,48
-30	-30,50
-40	-40,41
-50	-50,11
-60	-60,74
-70	-70,74
-80	-81,01
-90	-83,02

Setelah melewati sudut 90° sudut akan kembali turun dan diukur terhadap

sumbu z negatif. Kekurangan pada pembacaan data ini tidak mempengaruhi kinerja robot karena robot hanya bekerja pada sudut -90° sampai dengan 90°.

B. Pengujian Kontrol PWM

Hasil pengujian kontrol motor ditunjukkan pada Tabel 4.

Tabel 4 Hasil pengujian kontrol motor

Roda	Nilai PWM	Arah Putar	Kecepatan Putar (rpm)
Kanan	50	Maju	91,6
	100	Maju	130,5
	150	Maju	148,4
	200	Maju	156,6
	250	Maju	160,7
	255	Maju	162,0
	-50	Mundur	95,2
	-100	Mundur	138,8
	-150	Mundur	155,5
	-200	Mundur	163,0
	-250	Mundur	169,8
	-255	Mundur	170,0
Kiri	50	Maju	72,0
	100	Maju	136,6
	150	Maju	156,6
	200	Maju	166,4
	250	Maju	172,5
	255	Maju	175,4
	-50	Mundur	80,0
	-100	Mundur	138,1
	-150	Mundur	157,8
	-200	Mundur	167,1
	-250	Mundur	176,7
	-255	Mundur	177,2

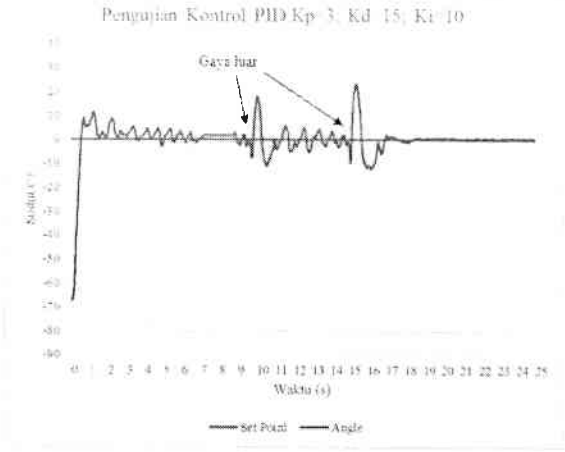
Pengujian kontrol motor dilakukan dengan mengukur kecepatan putar motor pada nilai PWM yang berbeda. Pengukuran kecepatan putar motor diukur menggunakan *digital tachometer* dan ditunjukkan dalam satuan *revolutions per minute* (rpm). Nilai PWM positif untuk arah putar motor ke depan dan nilai PWM negatif untuk arah putar motor ke belakang.

Dari hasil pengujian kontrol motor didapatkan bahwa kontrol arah putar motor sudah sesuai dengan arah putar motor yang diharapkan. Kecepatan putar motor hampir sama untuk nilai PWM yang sama, dan memiliki kecepatan putar yang semakin tinggi dengan nilai absolut dari PWM yang semakin besar.

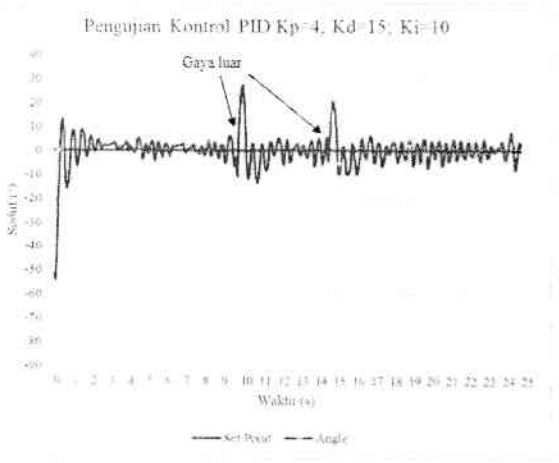
C. Pengujian Kontrol PID

Pengujian kontrol PID dilakukan dengan membandingkan respons robot saat diberikan nilai Kp, Ki, dan Kd yang berbeda-beda. Nilai awal Kp, Ki, dan Kd diambil dari nilai yang cukup stabil saat dilakukan *tuning* PID awal, yaitu Kp=3;Kd=15;Ki=10. Kemudian akan diambil data respons robot saat nilai Kp dinaikan dari nilai awal. Hal yang sama juga dilakukan untuk nilai Kp diturunkan, nilai Ki dinaikan, nilai Ki diturunkan, nilai Kd dinaikan, dan nilai Kd diturunkan dari nilai awal. Respons robot dilihat dari nilai sudut robot, respons yang baik adalah saat robot dapat memperbaiki sudutnya kembali menuju *set point* dengan cepat dan mempertahankan sudut pada *set point*. Dalam pengujian kontrol PID, robot diberi sudut awal sekitar -60° kemudian dibuat grafik sudut kemiringan robot terhadap waktu, dengan sudut dalam derajat dan waktu dalam sekon. Kemudian pada detik ke-10 dan detik ke-15 pengukuran diberi gaya dari luar pada robot. Grafik hasil pengujian kontrol PID untuk nilai Kp, Kd, dan Ki awal ditunjukkan pada Gambar 9.

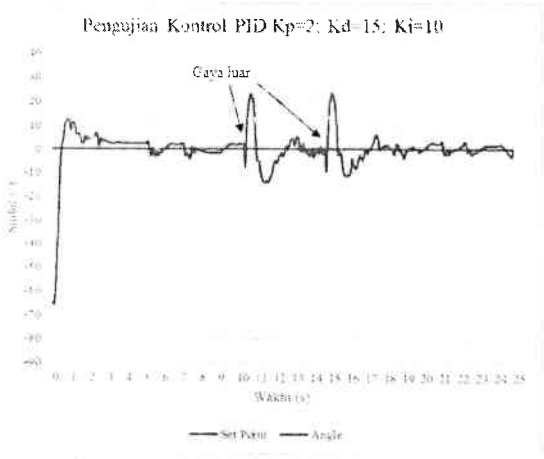
Untuk mengetahui pengaruh perubahan nilai Kp terhadap respons robot, dilakukan pengujian dengan penambahan dan pengurangan nilai Kp pada nilai Kd dan Ki tetap pada nilai awal. Grafik hasil pengujian kontrol PID setelah dilakukan penambahan nilai Kp ditunjukkan pada Gambar 10, dan grafik hasil pengujian kontrol PID setelah dilakukan pengurangan nilai Kp ditunjukkan pada Gambar 11.



Gambar 9 Grafik pengujian kontrol PID dengan nilai Kp=3; Kd=15; Ki=10



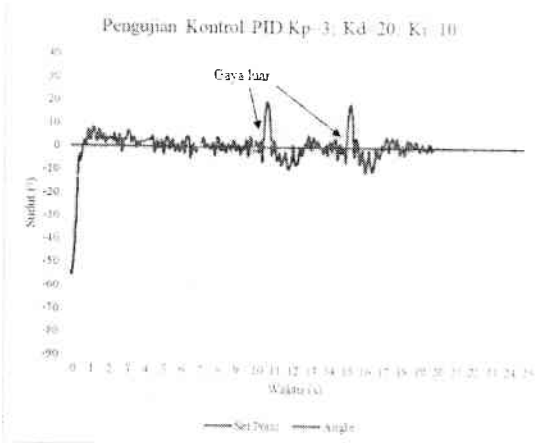
Gambar 10 Grafik pengujian kontrol PID dengan nilai Kp=4; Kd=15; Ki=10



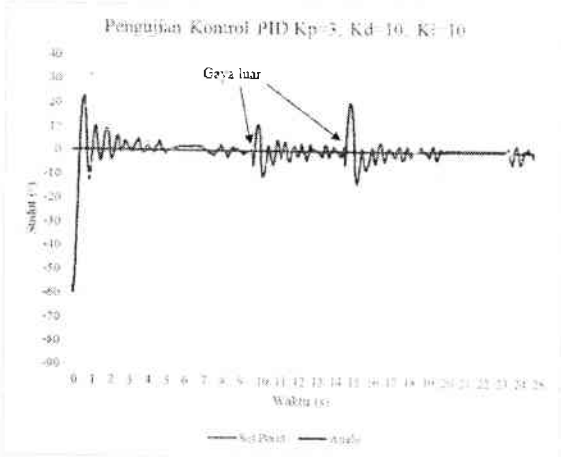
Gambar 11 Grafik pengujian kontrol PID dengan nilai Kp=2; Kd=15; Ki=10

Nilai Kp mempengaruhi kecepatan motor terhadap perbedaan sudut atau *error*. Dari hasil pengujian didapatkan bahwa nilai Kp yang terlalu besar akan mempercepat sudut kembali pada *set point*, tetapi menimbulkan osilasi dan tidak mencapai nilai yang stabil. Sedangkan untuk nilai Kp yang kecil robot dapat mencapai kestabilan, tetapi lebih lambat untuk mencapai *set point*. Karena pada titik stabil robot masih dapat jatuh, maka tetap terjadi perubahan nilai sudut yang direspons lambat oleh motor.

Untuk mengetahui pengaruh perubahan nilai Kd terhadap respons robot, dilakukan pengujian dengan penambahan dan pengurangan nilai Kd pada nilai Kp dan Ki tetap pada nilai awal. Grafik hasil pengujian kontrol PID setelah dilakukan penambahan nilai Kd ditunjukkan pada Gambar 12, dan grafik hasil pengujian kontrol PID setelah dilakukan pengurangan nilai Kd ditunjukkan pada Gambar 13.



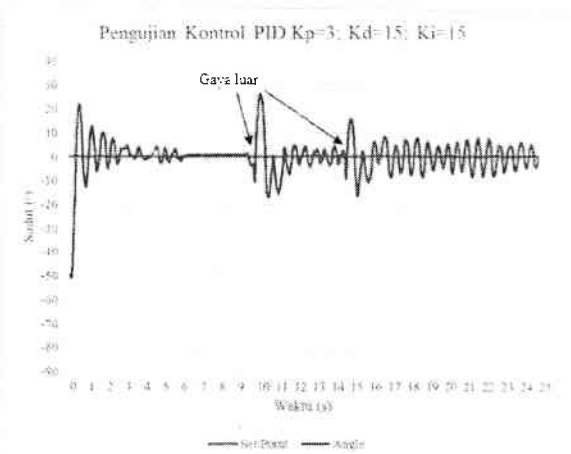
Gambar 12 Grafik pengujian kontrol PID dengan nilai Kp=3; Kd=20; Ki=10



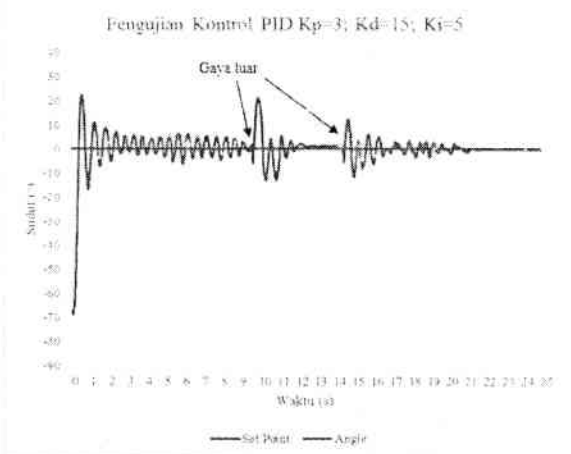
Gambar 13 Grafik pengujian kontrol PID dengan nilai Kp=3; Kd=10; Ki=10

Nilai Kd mempengaruhi kecepatan motor terhadap perbedaan *error*. Dari hasil pengujian didapatkan bahwa nilai Kd yang terlalu kecil tidak dapat menahan kecepatan motor saat mengembalikan nilai sudut pada *set point* sehingga menimbulkan *over shoot*. Sedangkan untuk nilai Kd yang lebih besar kecepatan robot dapat dikurangi saat mendekati *set point* dan mengurangi *over shoot*, tetapi saat nilai Kd terlalu besar kecepatan motor akan ditambahkan pada saat ada penambahan *error* dan dapat menyebabkan osilasi.

Untuk mengetahui pengaruh perubahan nilai Ki terhadap respons robot, dilakukan pengujian dengan penambahan dan pengurangan nilai Ki pada nilai Kp dan Kd tetap pada nilai awal. Grafik hasil pengujian kontrol PID setelah dilakukan penambahan nilai Ki ditunjukkan pada Gambar 14, dan grafik hasil pengujian kontrol PID setelah dilakukan pengurangan nilai Ki ditunjukkan pada Gambar 15.



Gambar 14 Grafik pengujian kontrol PID dengan nilai Kp=3; Kd=15; Ki=15



Gambar 15 Grafik pengujian kontrol PID dengan nilai Kp=3; Kd=15; Ki=5

Nilai Ki mempengaruhi kecepatan motor terhadap nilai *error* kumulatif. Nilai Ki yang terlalu besar akan menyebabkan sulitnya membalik arah motor saat tercapai nilai stabil karena nilai kumulatif *error* harus melewati *set point* agar dapat berkurang. Selain itu juga terjadi penambahan kecepatan motor dalam jangka panjang dari nilai *error* yang kecil, sehingga dapat menggerakan motor pada saat kondisi stabil. Nilai Ki yang terlalu kecil akan menimbulkan respons motor yang lambat untuk mengembalikan sudut pada *set point*, terutama saat ada perbedaan sudut yang besar. Kemudian setelah dilakukan *tuning* lebih lanjut, didapatkan nilai yang paling stabil untuk Kp, Kd, dan Ki, yaitu Kp=4; Kd=25; Ki=40.

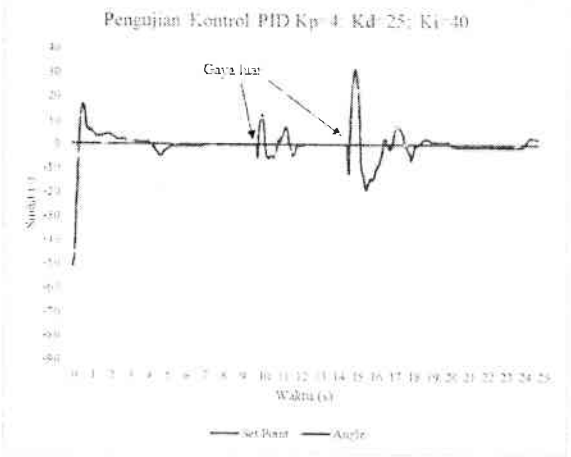
SIMPULAN DAN SARAN

Dari penelitian ini dapat disimpulkan bahwa:

a. MPU-6050 sebagai sensor *gyroscope* dan *accelerometer* dapat digunakan untuk menunjukkan besar sudut kemiringan robot setelah diproses dan difilter dengan *complementary filter* sehingga dapat dipergunakan sebagai sensor pada *two wheels self-balancing robot.*
b. Arah dan kecepatan pergerakan motor dapat dikendalikan menggunakan sinyal PWM yang lebar pulsanya diperoleh dari perhitungan nilai PID.
c. Kontrol PID dapat digunakan untuk kontrol *two wheels self-balancing robot* untuk menjaga keseimbangan dirinya sendiri dengan mengatur nilai Kp, Ki dan Kd yang sesuai. Untuk robot yang dirancang dalam penelitian ini, nilai Kp, Ki dan Kd yang sesuai adalah Kp=4; Kd=25; Ki=40.

Untuk penelitian selanjutnya, dapat dilakukan pengembangan berupa:

a. Robot diberikan kontrol eksternal agar dapat digerakkan maju, mundur, dan atau berbelok arah.
b. Robot dapat dibuat dalam ukuran lebih besar, sehingga dapat dinaiki oleh manusia.



Gambar 16 Grafik pengujian kontrol PID dengan nilai Kp=4; Kd=25; Ki=40

DAFTAR PUSTAKA

[1] Arduino. *Arduino Nano*. (https://www. arduino.cc/en/Main/ArduinoBoardNano; diakses 18 Februari 2016).

[2] Gunawan, A. P., Subagiyo, H., dan Wahyuni, R. T. 2013. Kontrol Kesetimbangan pada Robot Beroda Dua Menggunakan Pengendali PID dan *Complementary Filter*. *Jurnal Teknik Elektro dan Komputer*, 1(1): 1-11.

[3] InvenSense. *MPU-6000 and MPU-6050 Product Specification Revision 3.4*. (http://43zrtwysvxb2gf2 9r5o0athu.wpengine.netdna-cdn.com/ wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf; diakses 24 April 2016).

[4] Maa, M., Rahmawaty, M., & Ketaren, L. P. Balancing Robot Beroda Dua Menggunakan Metoda Kontrol Proporsional, Integral dan Derivatif. *Jurnal Elementer*, 1(2): 39-48.

[5] Ooi, Rich Chi. 2003. *Balancing a Two-Wheeled Autonomous Robot*. Tesis. Perth: The University of Western Australia School of Mechanical Engineering,

[6] Rachana, S., Latha, B. N., & C, S. S. 2016. Sensor Data Acquisition Methodology using Complementary Filter with IMU Sensor. *Internatonal Journal of Information Technology and ComputerEngineering*, 1(2): 36-41.

[7] Texas Instrument. *L293x Quadruple Half-H Drivers*. (http://www.ti.com/lit/ds/symlink/l293. pdf; diakses 24 April 2016).

[8] Yoo, T. S., Hong, S. K., Yoon, H. M., & Park, S. 2011. Gain-Scheduled Complementary Filter Design for a MEMS Based Attitude and Heading Reference System. *Sensors*, 11(4): 3816-3830.