

**IMPLEMENTASI PID CONTROLLER PADA *SELF
BALANCING ROBOT***

NASKAH PUBLIKASI TUGAS AKHIR



ADETIYA PRATAMA

5140711013

**PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INFORMASI DAN ELEKTRO
UNIVERSITAS TEKNOLOGI YOGYAKARTA**

**YOGYAKARTA
2018**

**HALAMAN PENGESAHAN
NASKAH PUBLIKASI TUGAS AKHIR MAHASISWA**

Judul Tugas Akhir:
IMPLEMENTASI PID *CONTROLLER* PADA *SELF BALANCING ROBOT*

Judul Naskah Publikasi:
IMPLEMENTASI PID *CONTROLLER* PADA *SELF BALANCING ROBOT*

Disusun oleh:
ADETIYA PRATAMA
5140711013

Mengetahui

Nama	Jabatan	Tanda Tangan	Tanggal
Dr. Arief Hermawan, S.T., M.Eng.	Pembimbing

Naskah Publikasi Tugas Akhir ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana pada Program Studi Teknik Elektro

Yogyakarta,2018
Ketua Program Studi Teknik Elektro

Satyo Nuryadi, S.T., M.Eng.
NIK. 100205023

PERNYATAAN PUBLIKASI

Yang bertanda tangan di bawah ini, Saya:

Nama : Adetiya Pratama
NIM : 5140711013
Program Studi : Teknik Elektro
Fakultas : Teknologi Informasi dan Elektro

“Implementasi PID Controller Pada Self Balancing Robot”

Menyatakan bawa Naskah Publikasi ini hanya akan dipublikasikan di JURNAL FTIE UTY dan tidak akan dipublikasikan di jurnal yang lain.

Demikian surat pernyataan ini dibuat dengan sebenar-benarnya.

Yogyakarta, 31 Juli 2018

Penulis,

Adetiya Pratama

5140711013

IMPLEMENTASI PID CONTROLLER PADA SELF BALANCING ROBOT

ADETIYA PRATAMA

Program Studi Teknik Elektro, Fakultas Teknologi Informasi dan Elektro
Universitas Teknologi Yogyakarta
Jl. Ringroad Utara Jombor, Sleman, Yogyakarta
E-mail: aditsloow@gmail.com

ABSTRAK

Self Balancing Robot merupakan salah satu penerapan sistem pendulum terbalik dimana sistem tersebut memiliki gravitasi yang berada diatas poros putar sehingga menyebabkan sistem kurang stabil dan tidak seimbang. Untuk itu, diperlukan suatu sistem kendali yang dapat membuat pendulum seimbang dengan menggerakkan dua buah roda yang menjadi poros putar dari pendulum terbalik. Tujuan dari penelitian ini adalah merancang suatu sistem pengendalian robot beroda dua agar dapat mempertahankan posisi dalam keadaan seimbang. Sistem ini mendapatkan masukan dari sensor MPU-6050 yaitu gabungan antara akselerometer dan giroskop. Sistem pengendalian yang digunakan yaitu metode kendali PID (Proportional Integral dan Derivative). Proses kendali PID ini dilakukan dengan membuat suatu kode program menggunakan software Arduino IDE yang nantinya dimasukkan ke Arduino Uno yang hasilnya digunakan untuk mengontrol kecepatan dan arah putaran motor DC, serta set point nilai kemiringan sudut dari robot. Motor DC akan berputar ke depan dan belakang apabila sudut yang diperoleh lebih dari nilai set point. Nilai konstanta PID yang didapat berdasarkan tuning trial and error.

Kata Kunci: *Self Balancing Robot, PID, MPU-6050*

ABSTRACT

Self Balancing Robot is one application of inverted pendulum system where the system has a gravity above the rotary shaft causing the system is less stable and unbalanced. Therefore, we need a control system that can make the pendulum balanced by moving two wheels into a rotary shaft of the inverted pendulum. The purpose of this research is to design a two-wheeled robot control system in order to maintain the position in a balanced state. This system gets input from the sensor MPU-6050 is a combination of accelerometer and gyroscope. Control system used is PID control method (Proportional Integral and Derivative). PID control process is done by creating a program code using Arduino IDE software which will be entered into Arduino Uno which the result is used to control the speed and direction of DC motor rotation, and set point value of the angle of the robot. The DC motor will spin forward and back when the angle obtained is more than the set point value. The value of PID constants obtained based on trial and error tuning.

Keywords: *Self Balancing Robot, PID, MPU-6050*

Nama	Jabatan	Tanda Tangan	Tanggal
Satyo Nuryadi, S.T., M.Eng.	Ketua Program Studi Teknik Elektro		

1. PENDAHULUAN

Perkembangan teknologi robotika telah membuat kualitas kehidupan manusia semakin tinggi. Saat ini, perkembangan teknologi robotika telah mampu meningkatkan kualitas maupun kuantitas berbagai industri. Salah satu cara menambah tingkat kecerdasan sebuah robot adalah dengan menambah sensor, metode kontrol, bahkan memberikan kecerdasan buatan pada robot tersebut. Salah satunya adalah *balancing robot*.

Balancing robot beroda dua merupakan suatu robot yang memiliki prinsip kerja seperti pendulum terbalik yang mempertahankan keseimbangan robot dan tegak lurus terhadap permukaan bumi di bidang datar. Pendulum terbalik adalah pendulum yang terengsel ke kereta beroda yang dapat bergerak maju dan mundur pada bidang horisontal di sepanjang lintasan. Penerapan konsep *inverted pendulum* dalam dunia robotika dapat dilihat pada *balancing robot*, yaitu *mobile robot* dengan dua roda yang mana roda tersebut diasumsikan sebagai kereta beroda dan badan robot diasumsikan sebagai pendulum. Sistem ini tidak stabil karena ketika kereta beroda diberi gangguan dari luar maka pendulum akan terjatuh. Dibutuhkan sebuah sistem kontrol yang sesuai untuk mengontrol sistem sehingga seimbang dan stabil. Tujuan utama dari proyek ini adalah dengan menggunakan strategi kontrol yang baik untuk menjaga robot dalam keadaan seimbang pada bidang datar.

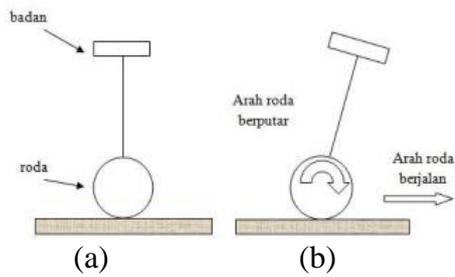
Alogaritma yang umum digunakan sebagai pengendali keseimbangan *balancing robot* adalah kendali PID (*Proportional Integral Derivative*). Algoritma ini terdiri dari bagian kendali proporsional, integral dan derivatif. Kelebihan dari teknik kendali ini adalah pada

fleksibilitasnya untuk dapat diterapkan pada berbagai macam sistem kendali. Kendali PID terbukti paling sesuai diterapkan untuk mengendalikan suatu sistem yang tidak diketahui pemodelan matematikanya, sehingga metode desain kendalinya tidak dapat dicari secara analitis. Kekurangannya adalah, karena penerapannya yang secara umum berdasarkan hubungan masukan-keluaran, maka hasil yang diperoleh belum tentu merupakan kendali yang optimal. Masalah yang didapatkan adalah merealisasikan metode kendali PID pada sistem robot, sehingga robot ini dapat mempertahankan posisinya tegak lurus dengan seimbang terhadap permukaan bumi pada bidang datar.

2. TINJAUAN PUSTAKA

2.1 *Balancing Robot*

Balancing robot beroda dua merupakan *mobile robot* yang memiliki dua buah roda dan tidak akan seimbang tanpa adanya kontroler (Bimarta dkk., 2015). *Balancing robot* ini merupakan pengembangan dari model pendulum terbalik (*inverted pendulum*) yang diletakkan di atas kereta beroda. Permasalahannya adalah ketika kereta beroda bergerak maju dan mundur, maka pendulum yang ujungnya menyatu pada kereta beroda akan jatuh. Oleh karena itu, dibutuhkan sebuah kendali sehingga pendulum tetap seimbang dan berdiri tegak. Prinsip kerja dari *balancing robot* ditunjukkan pada Gambar 2.2.



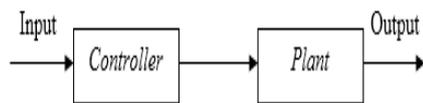
Gambar 2.2 Prinsip Kerja *Balancing Robot*

2.2 Sistem Kendali

Sistem kendali terdiri dari *sub system* dan proses (*plant*) yang dibangun untuk mendapatkan keluaran yang diinginkan sesuai dengan nilai *set point*. Dalam dasar sistem kendali, terdapat dua sistem konfigurasi utama yang sering digunakan, yaitu sistem *Open Loop* dan *Close Loop*.

a. Sistem Kontrol Loop Terbuka (*Open-Loop Control System*)

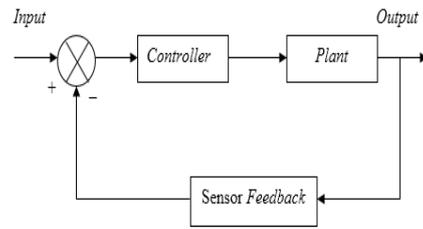
Suatu sistem kontrol yang mempunyai karakteristik dimana nilai keluaran tidak memberikan pengaruh pada aksi kontrol disebut sistem kontrol *loop* terbuka (Triwiyatno, 2011).



Gambar 2.3 Sistem Kontrol Loop Terbuka

b. Sistem Kontrol Loop Tertutup (*Closed-Loop Control System*)

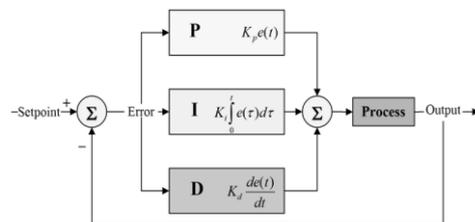
Sistem kontrol *loop* tertutup adalah identik dengan sistem kontrol umpan balik, dimana nilai dari keluaran akan ikut mempengaruhi pada aksi kontrolnya (Triwiyatno, 2011).



Gambar 2.4 Sistem Kontrol Loop Tertutup

2.3 Kendali PID (*Proportional Integral Derivative*)

Kendali PID adalah sistem kendali yang susunan komponen fisik dihubungkan sedemikian rupa untuk mengatur suatu kondisi agar mencapai kondisi yang diharapkan (Odinanto dkk., 2015). Sistem kendali ini terdiri dari tiga bagian, yaitu input, proses, dan output. Ketiganya dapat dipakai bersamaan maupun sendiri-sendiri tergantung dari respon yang kita inginkan terhadap suatu *plant*. Blok diagram sederhana dari kontrol PID dapat dilihat pada Gambar 2.5.



Gambar 2.5 Blok Diagram Pengendali PID

2.4 Motor DC

Motor DC (*Direct Current*) adalah peralatan elektromagnetik dasar yang berfungsi untuk mengubah energi listrik menjadi energi mekanik (Hidayati, 2014). Motor DC dikendalikan dengan menentukan arah dan kecepatan putarnya. Kecepatan putar motor DC diatur dengan besarnya arus yang diberikan.



Gambar 2.6 Motor DC

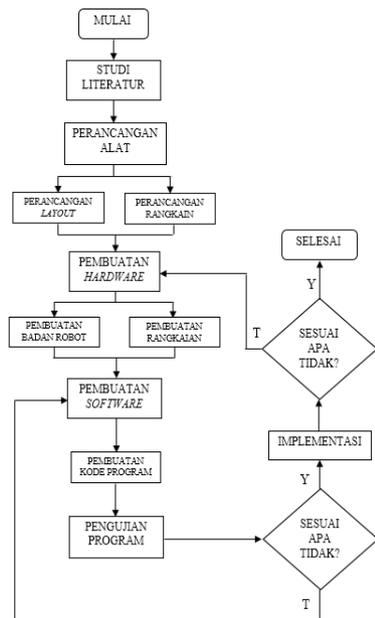
2.5 Sensor MPU

MPU adalah *chip IC inverse* yang didalamnya terdapat sensor *accelerometer* dan *gyroscope* yang sudah terintegrasi (Firman, 2017). *Accelerometer* berfungsi untuk mengukur percepatan akibat gravitasi dan mendeteksi getaran (vibrasi). Sedangkan *gyroscope* berfungsi untuk mengukur atau menentukan orientasi suatu benda berdasarkan pada ketepatan momentum sudut.



Gambar 2.7 Sensor MPU

3. METODE PENELITIAN



Gambar 3.1 Flowchart Metode Penelitian

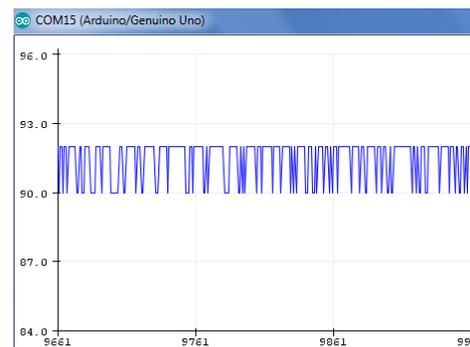
Dari Gambar 3.1 *flowchart* metode penelitian, tahapan atau rencana penelitian yang akan digunakan sebagai berikut: (1) Studi Literatur, (2) Perancangan Alat, (3) Pembuatan Alat, (4) Pembuatan Kode Program, (5) Pengujian Program, (6) Implementasi, (7) Penyusunan Laporan.

4. PERANCANGAN ALAT

4.1 Perancangan Sistem PID

Metode yang digunakan untuk mendapatkan nilai PID merupakan metode Ziegler-Nichols dengan *tuning* osilasi. Langkah-langkah untuk melakukan *tuning* PID menggunakan metode ini adalah sebagai berikut:

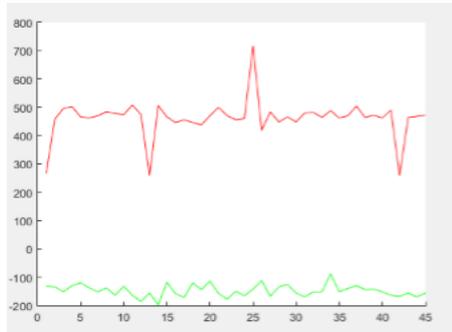
- Ki dan kd diberi nilai 0
- Nilai kp dinaikkan terus sampai sistem mengalami osilasi yang stabil pada *serial plotter*.
- Gambar 4.1 merupakan hasil percobaan kalibrasi sensor, kemudian diperoleh gelombang osilasi dengan magnitudo sama berdasarkan nilai kp dengan *range* 112-120. Nilai ku (*ultimate gain*) yang didapatkan sebesar 116 yaitu rata-rata dari nilai kp tersebut.



Gambar 4.1 Gelombang Osilasi Hasil Kalibrasi Sensor

- Penentuan *pu* (*ultimate periode*) dilakukan berdasarkan jarak waktu antar maksimum *overshoot* dengan menggunakan matlab. Pengamatan dilakukan terhadap sumbu y (sinyal

hijau) antar maksimum *overshoot* pada titik 5 dan titik 10 dengan waktu 0.5 s (500 ms). Waktu antara titik 5 dan titik 10 tersebut di tulis sebagai nilai pu (pu = 500).



Gambar 4.2 Hasil Eksperimen Menentukan Nilai Pu

- e. Menghitung nilai k_p , k_i , dan k_d Model penalaan PID dengan metode Ziegler-Nichols *tuning* osilasi dapat dilihat pada Tabel 4.2.

Tabel 4.2 *Tuning* PID Metode Osilasi

Tipe Kontrol	K_p	K_i	K_d
P	$0.5K_u$	0	0
PI	$0.45K_u$	$\frac{P_u}{1.2}$	-
PID	$0.6K_u$	$0.5P_u$	$0.125P_u$

Berdasarkan Tabel 4.2, maka didapatkan hasil sebagai berikut:

$$K_p = 0.6 \cdot K_u = 0.6 \cdot 116 = 69.6$$

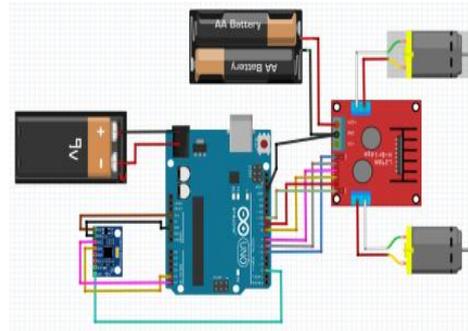
$$K_i = 0.5 \cdot P_u = 0.5 \cdot 500 = 250$$

$$K_d = 0.125 \cdot P_u = 0.125 \cdot 500 = 62.5$$

Nilai k_p , k_i dan k_d yang diperoleh selanjutnya akan di-*embedd*-kan ke dalam sistem *balancing robot*.

4.2 Rangkaian Sistem

Rangkaian dari sistem *Self Balancing Robot* dapat dilihat pada Gambar 4.3.

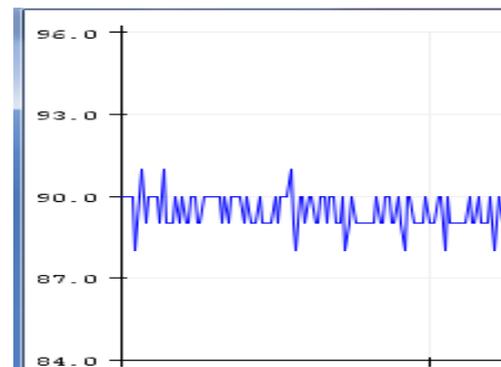


Gambar 4.3 Rangkaian Sistem Robot.

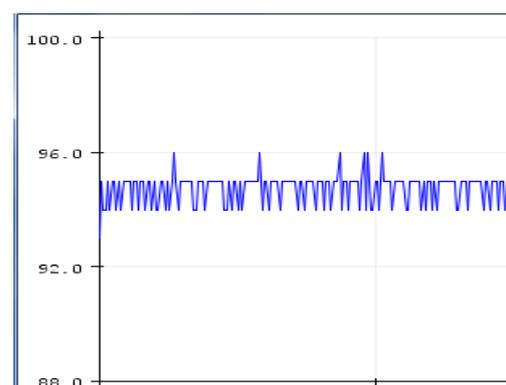
5. HASIL DAN PEMBAHASAN

5.1 Pengujian Sensor

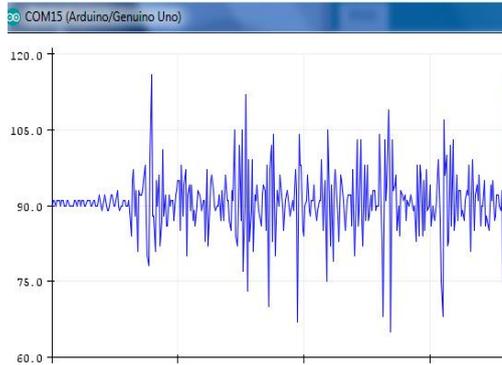
Pengujian ini dilakukan untuk mengetahui kepekaan sensor terhadap sistem sesuai dengan yang diinginkan.



Gambar 5.1 Respon Sensor Sudut 180° Saat Diam



Gambar 5.2 Respon Sensor Sudut 175° Saat Diam

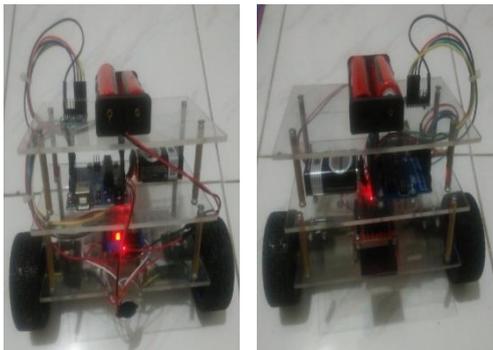


Gambar 5.3 Respon Sensor Saat Robot Bergerak

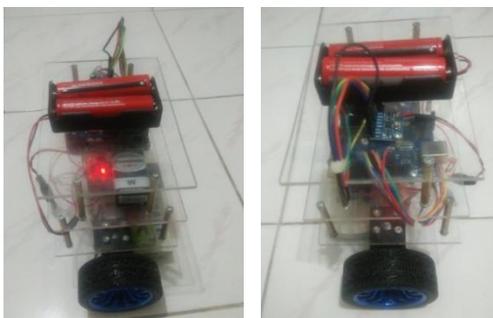
Dari hasil pengujian dapat disimpulkan bahwa sensor bekerja dengan baik

5.2 Hasil Rancangan Robot

Hasil rancangan perangkat keras dari *Self Balancing Robot* dapat dilihat pada Gambar 5.4 dan Gambar 5.5.



Gambar 5.4 Robot Tampak Depan Dan Belakang



Gambar 5.5 Robot Tampak Samping Kiri Dan Kanan

5.3 Hasil Percobaan Robot

Proses kendali ini diprogram menggunakan Arduino IDE yang diumpankan ke motor DC untuk mengatur kecepatan dan arah putaran motor dan kemudian menentukan posisi keseimbangan robot. Selanjutnya, sensor akan bekerja mendeteksi sudut kemiringan robot agar tidak terjatuh baik dalam keadaan maju maupun mundur. Karena metode yang digunakan dalam menentukan *set point* nilai kemiringan sudut menggunakan metode manual *tunning*, maka nilai yang digunakan sebesar $175,50^{\circ}$.

Untuk menentukan nilai dari parameter tersebut, hal pertama yang dilakukan adalah dengan memasukkan nilai k_p , k_i , dan k_d sama dengan nol, kemudian sesuaikan nilai k_p . Terlalu sedikit k_p akan membuat robot terjatuh (tidak cukup koreksi), terlalu banyak k_p yang akan membuat robot bolak-balik dengan liar. k_p yang cukup baik akan membuat robot sedikit bolak-balik. Setelah k_p diset, atur k_d . Nilai k_d yang bagus akan mengurangi osilasi sampai robot hampir stabil. Jumlah k_d yang tepat akan membuat robot bisa berdiri dan berjalan meski masih terjatuh. Terakhir, mengatur nilai parameter k_i . Robot akan beresilasi saat dinyalakan bahkan jika k_p dan k_d ditetapkan, namun akan stabil pada waktunya. Nilai k_i yang tepat akan mempersingkat waktu yang dibutuhkan robot untuk dapat menyeimbangkan diri. Data hasil percobaan untuk menentukan nilai k_p , k_i , dan k_d dapat dilihat pada Tabel 5.1.

Tabel 5.1 Hasil Percobaan Untuk Menentukan Nilai Kp, Ki, dan Kd

No	Nilai Kp, Ki, Kd	Respon
1	Kp = 0 Ki = 0 Kd = 0	Robot tidak bergerak sama sekali
2	Kp = 69.6 Ki = 0 Kd = 0	Robot belum dapat berdiri seimbang dan bergerak bolak-balik secara liar
3	Kp = 69.6 Ki = 250 Kd = 0	Robot dapat berdiri dan bergerak bolak-balik secara liar, tetapi masih jatuh
4	Kp = 69.6 Ki = 0 Kd = 62.5	Robot dapat berdiri dan berjalan tetapi masih jatuh
5	Kp = 0 Ki = 250 Kd = 62.5	Robot bergerak, tetapi belum dapat berdiri
6	Kp = 69.6 Ki = 250 Kd = 62.5	Robot dapat berdiri seimbang

Berdasarkan hasil percobaan, maka nilai yang memiliki respon yang baik terhadap keseimbangan robot yaitu $k_p = 69.6$, $k_i = 250$, dan $k_d = 62.5$. Setelah mendapatkan nilai parameter k_p , k_i , dan k_d , kemudian dilakukan pengujian robot dengan nilai *set point*, konstanta k_p , k_i , k_d , serta putaran motor yang berbeda-beda maka

didapat hasil seperti Tabel 5.2, Tabel 5.3, dan Tabel 5.4.

Tabel 5.2 Hasil Percobaan Robot Dengan Nilai *Set Point* Berbeda-Beda

<i>Set Point</i>	Kp	Ki	Kd	Respon
175.50 ⁰	69.6	250	62.5	Robot dapat seimbang
174.50 ⁰	69.6	250	62.5	Robot dapat seimbang dalam beberapa detik
173.50 ⁰	69.6	250	62.5	Tidak Seimbang
172.50 ⁰	69.6	250	62.5	Tidak Seimbang
171.50 ⁰	69.6	250	62.5	Tidak Seimbang

Berdasarkan hasil percobaan yang telah dilakukan, ketika nilai *set point* diatur berkisar 171.50⁰ sampai 174.50⁰, maka respon yang terjadi pada robot mengalami keadaan tidak seimbang. Robot dalam keadaan

seimbang ketika nilai *set point* diatur 175.50^0 . Pengaturan nilai *set point* kemiringan sudut yang berbeda mengindikasikan bahwa kondisi robot mempunyai karakteristik respon yang berbeda.

Tabel 5.3 Hasil Percobaan Robot Dengan Nilai Konstanta K_p , K_i , K_d Berbeda-Beda

<i>Set Point</i>	K_p	K_i	K_d	Respon
175.50	76	184	69	Robot dapat seimbang, tetapi masih ada getaran kecil
175.50	80	190	76	Robot dapat seimbang, tetapi masih ada getaran kecil
175.50	82	178	57	Robot dapat seimbang, tetapi masih ada getaran lumayan besar
175.50	78	190	50	Robot dapat seimbang, tetapi masih ada getaran cukup besar
175.50	100	300	55	Robot dapat seimbang, tetapi masih ada getaran cukup besar

Berdasarkan hasil percobaan dengan nilai *set point* yang konstan dan nilai parameter PID yang berbeda, diperoleh kondisi robot mengalami keadaan seimbang dengan karakteristik keadaan yang berbeda-beda. Selanjutnya dilakukan pengujian untuk menentukan kisaran nilai *setpoint* yang masih bisa mempertahankan robot agar tidak jatuh.

Tabel 5.4 Hasil Pengujian Menentukan Kisaran Nilai *Set Point* Yang Stabil

<i>Set Point</i>	K_p	K_i	K_d	Respon
175^0	69.6	250	62.5	Robot tidak seimbang
175.10^0	69.6	250	62.5	Robot tidak seimbang
175.20^0	69.6	250	62.5	Robot tidak seimbang
175.30^0	69.6	250	62.5	Robot dapat seimbang
175.40^0	69.6	250	62.5	Robot dapat seimbang
175.50^0	69.6	250	62	Robot dapat seimbang
175.60^0	69.6	250	62.5	Robot dapat seimbang
175.70^0	69.6	250	62.5	Robot dapat seimbang
175.80^0	69.6	250	62.5	Robot tidak seimbang
175.90^0	69.6	250	62.5	Robot tidak seimbang

Dari hasil pengujian, robot dapat seimbang atau tidak jatuh dengan kemiringan pada kisaran nilai *setpoint* 175.30^0 - 175.70^0 . Dengan penyesuaian dari ketiga nilai parameter kendali,

pengaruh PID pada sistem dapat dijabarkan sebagai berikut:

1. Proporsional (P)

Kontrol proporsional memiliki sifat sebagai faktor penguat, artinya:

- a. Jika nilai K_p besar, maka kecepatan respon mencapai nilai *set point* meningkat.
- b. Jika nilai K_p kecil, maka kecepatan respon melambat untuk mencapai nilai *set point* sehingga robot akan jatuh.
- c. Jika nilai K_p besar melewati batas normal, maka sistem akan berosilasi dan membuat sistem tidak stabil.

2. Integral (I)

- a. Kontrol integral memiliki respon lebih lambat, artinya jika nilai K_i tidak tepat dapat menyebabkan respon (waktu) yang dibutuhkan untuk mencapai nilai *set point* lebih lama, sehingga dapat menyebabkan sistem tidak stabil.
- b. Kontrol integral dapat memperbaiki *overshoot*.
- c. Memperbaiki waktu yang dibutuhkan untuk mencapai keadaan yang menyatakan posisi stabil (*steady state*).

3. Derivatif (D)

- a. Sifat dari kontrol derivatif ini dapat memberikan koreksi kesalahan.
- b. Memperbaiki respon (waktu) untuk mencapai nilai *set point*.
- c. Memberikan efek redaman pada sistem yang berosilasi.

6. PENUTUP

6.1 Kesimpulan

Berdasarkan hasil penelitian setelah dilakukannya pengujian terhadap *Self Balancing Robot* yang telah dibuat,

dapat disimpulkan beberapa poin sebagai berikut:

- a. *Self Balancing Robot* merupakan suatu sistem yang tidak stabil tanpa adanya kendali.
- b. Sistem kendali PID yang diterapkan pada *balancing robot* ini bekerja dengan cukup baik dan dapat mengendalikan robot agar tetap seimbang dengan nilai dari parameter $K_p = 69.6$, $K_i = 250$ dan $K_d = 62.5$.
- c. *Balancing robot* yang dibuat mampu menyeimbangkan diri dengan baik dengan kemiringan sudut 175.50° terhadap bidang datar saat robot dijalankan.

6.2 Saran

Untuk pengembangan lebih lanjut pada sistem *Self Balancing Robot* ini disarankan untuk melakukan hal-hal sebagai berikut:

- a. Metode *tunning* konstanta kendali PID dilakukan dengan metode *auto tunning* agar hasil yang diperoleh lebih optimal.
- b. Sistem *Self Balancing Robot* yang dibuat hanya dapat mempertahankan posisi robot dalam keadaan seimbang. Untuk selanjutnya, diharapkan robot dapat dikendalikan dengan menggunakan *remote controls*.
- c. Dalam pengembangan selanjutnya, diharapkan sistem ini dapat diterapkan secara langsung dengan membuat suatu alat transportasi.

DAFTAR PUSTAKA

- [1] Andriansyah, dan Hidyatama. (2013). *Rancang Bangun Prototype Elevator Menggunakan Mikrokontroler Arduino Atmega 328p*. Jakarta: Jurnal Teknologi Elektro

- Universitas Mercu Buana. Vol. 4, No. 3.
- [2] Bimarta, Eko Putra, dan Dharmawan. (2015). *Balancing Robot Menggunakan Metode Kendali Proporsional Integral Derivatif*. Yogyakarta: Jurnal Ilmu Komputer Dan Elektronika Universitas Gadjah Mada. Vol. 5, No. 1.
- [3] Bobby, Susanto, dan Suratman. (2015). *Implementasi Robot Keseimbangan Beroda Dua Berbasis Mikrokontroler*. Bandung: Jurnal ELKOMIKA Teknik Elektro Institut Teknologi Nasional. Vol. 3, No. 2.
- [4] Gunawan, Subagiyo, dan Wahyuni. (2013). *Kontrol Kesetimbangan Pada Robot Beroda Dua Menggunakan Pengendali PID Dan Complementary Filter*. Riau: Jurnal Teknik Elektro Dan Komputer. Vol. 1, No. 1.
- [5] Hidayati, Qory. (2014). *Pengaturan Kecepatan Putar Motor DC Dengan Menggunakan Mikrokontroler Atmega 8535*. Balikpapan: Jurnal Teknik Elektronika Politeknik Negeri Balikpapan.
- [6] Ketaren, Ma'a, dan Rahmawaty. (2015). *Balancing Robot Beroda Dua Menggunakan Metode Kontrol Proporsional, Integral, dan Derivatif*. Riau: Jurnal ELEMENTER Politeknik Caltex Riau. Vol. 1, No. 2.
- [7] Odianto, Suprijono, dan Widya Natasari. (2015). *Perancangan Pengendali Motor Arus Searah 1 HP 220 Volt Dengan Metode Pid Berbasis Mikrokontroler*. Surabaya: Jurnal Jurusan Teknik Elektro Institut Teknologi Adhi Tama Surabaya. Vol. 19, No. 1.
- [8] Ogata, Katsuhiko. (2010). *Modern Control Engineering Fifth Edition*. New York: PrenticeHall, Inc.
- [9] Raranda, dan Rusimamto. (2017). *Implementasi Kontroler PID Pada Two Wheels Self Balancing Robot Berbasis Arduino Uno*. Surabaya: Jurnal Teknik Elektro Universitas Negeri Surabaya. Vol. 6, No. 2.
- [10] Triwiyatno, Aris. (2011). *Buku Ajar Sistem Kontrol Analog*. Semarang: Universitas Diponegoro.
- [11] Wisesa, Tirta. (2014). *Perancangan Pengaturan Kecepatan Motor Induksi Satu Fasa Dengan Pwm Menggunakan Pengendali Pid Berbasis Arduino*. Bengkulu: Skripsi Program Studi Teknik Elektro Universitas Bengkulu.