



## *Balancing Robot Beroda Dua Menggunakan Metoda Kontrol Proporsional, Integral dan Derivatif*

Lio Prisko Ketaren<sup>1</sup>, Mustaza Ma'a<sup>2</sup> dan Made Rahmawaty<sup>3</sup>

<sup>1</sup>Politeknik Caltex Riau, email: [liopriskoketaren@gmail.com](mailto:liopriskoketaren@gmail.com)

<sup>2</sup>Politeknik Caltex Riau, email: [mustaza@pcr.ac.id](mailto:mustaza@pcr.ac.id)

<sup>3</sup>Universitas Gadjah Mada, email: [made.rahmawaty@mail.ugm.ac.id](mailto:made.rahmawaty@mail.ugm.ac.id)

### Abstrak

*Balancing robot beroda dua merupakan suatu robot yang memiliki prinsip kerja seperti pendulum terbalik yang mempertahankan keseimbangan robot pada sudut  $0^\circ$  dan tegak lurus terhadap permukaan bumi dibidang datar. Untuk kontrol keseimbangan balancing robot menggunakan metode kontrol Proporsional Integral Derivatif (PID) dengan mengatur kecepatan dan arah putaran motor. Balancing robot beroda dua menggunakan sensor accelerometer untuk mendeteksi kemiringan serta sensor gyroscope untuk mendeteksi kecepatan sudut badan robot ketika akan terjatuh. Untuk penggerak robot menggunakan motor DC. Balancing robot beroda dua mampu mempertahankan posisi robot pada kondisi setimbang. Dari hasil pengujian nilai PID yang paling ideal yaitu Kp: 8.0 Ki: 7.8 dan Kd 2.0*

**Kata kunci:** *Balancing robot, Arduino, Proporsional Integral Derivatif (PID), Accelerometer, Gyroscope*

### Abstract

*Two-wheeled balancing robot is a robot which has work principle like inverted pendulum that maintaining robot angle balance at  $0^\circ$  and it is perpendicular to flat surface of earth. Balancing robot control use Proportional Integral Derivatif (PID) control method included with adjusting motor speed and spinning direction. Two-wheeled balancing robot uses accelerometer sensor for detecting the tilt and gyroscope sensor for detecting body angle velocity when it falls. For robot movement, it uses DC motor. Two-wheeled balancing robot is able to maintain robot position at its balanced condition. In PID values result, the most ideal values are: Kp: 8.0 Ki: 7.8 and Kd 2.0*

**Keywords:** *Balancing Robot, Arduino, Proportional Integral Derivative (PID), Accelerometer, Gyroscope*

## 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. Teknologi robotika juga telah menjangkau sisi hiburan dan pendidikan bagi manusia. 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 beroda dua. Sistem *balancing* robot pertama

dideklarasikan oleh Dean Kamen tahun 2001 dengan nama SEGWAY yang kemudian dikenal sebagai "The first Self-balancing, electric powered transportation device".

Penelitian ini untuk menjelaskan design implementasi kesetimbangan robot beroda dua. Untuk pengontrolan digunakan Arduino, sensor *accelerometer* dan *gyroscope* serta kontrol Proporsional Integral Derivatif (PID) sebagai metode pengendali. *Accelerometer* digunakan untuk mendeteksi kemiringan dan *gyroscope* digunakan untuk mendeteksi kecepatan sudut badan robot ketika akan terjatuh.

## 2. Tujuan Penelitian

Tujuan dari penelitian ini adalah mempertahankan posisi robot pada kondisi seimbang dan tegak lurus terhadap permukaan bumi di bidang datar dengan menggunakan metode kontrol Proporsional Integral Derivatif (PID).

## 3. Metoda Penelitian

Menyeimbangkan robot kesetimbangan memerlukan suatu metode kontrol yang baik dan handal untuk mempertahankan posisi robot dalam posisi tegak lurus terhadap permukaan bumi [1]. Saat robot kesetimbangan condong atau miring ke kanan maka respon sistem adalah membuat motor memutar roda searah jarum jam sehingga robot kesetimbangan bergerak ke arah kanan. Gaya yang dihasilkan untuk menyeimbangkan berasal dari putaran roda yang berasal dari torsi yang dihasilkan oleh motor. Dalam penelitiannya, robot mencapai keadaan setimbang setelah mendapat gangguan saat dimiringkan dengan *tuning* parameter  $K_p=30$  dan  $T_i=0,2$ . Penelitian Robot tersebut menggunakan 2 buah sensor yaitu sensor *Accelerometer* dan *Gyroscope*, dan menggabungkan kedua *output* dengan menggunakan algoritma *Complementary Filter*. Pada penelitiannya mengatakan dapat digunakan metode kontrol lain, seperti kontrol *fuzzy*, Jaringan Saraf Tiruan (JST), algoritma genetik atau *Linear Quadratic Regulator* (LQR) untuk mengontrol keseimbangan badan robot terhadap permukaan bumi di bidang datar [1].

Peneliti lainnya yang juga menggunakan kontrol PID dan algoritma *Complementary Filter* pada *Balancing Robot* adalah Ade Putra Gunawan. Hal yang membedakannya adalah input yang dimasukkan menggunakan 2 buah konfigurasi, yaitu PI menggunakan sensor *Accelerometer* dan *Gyroscope*, dan PD menggunakan sensor *Rotary encoder*. Serta seting parameter yang didapat adalah  $K_p\_Speed = 1.5$ ;  $K_i\_Angle = 8$ ;  $K_p\_Rotation = 0.1$ ;  $K_d\_Speed = 0.2$  agar robot mencapai keadaan setimbang [2].

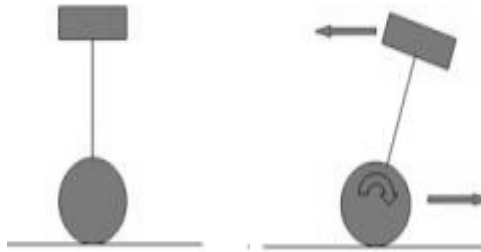
Dalam penelitian lainnya *Self Balancing Robot* (SBR) yang dilakukan oleh Handry Khoswanto dalam kesetimbangan robot beroda dua menggunakan metode *fuzzy logic* menghasilkan output kecepatan sudut. Pembacaan input dimodifikasi menjadi *Err\_Angle* dan *Del\_Err\_Angle*, hanya dengan menggunakan sensor *Accelerometer*. Kedua hasil tersebut digunakan sebagai *crisp input* dari blok *fuzzy Logic Controller*. Melalui proses fuzzifikasi dan defuzzifikasi akan menghasilkan *crisp output* berupa  $\omega$  (kecepatan sudut). Dari penelitian dengan 2 buah *input* pada proses fuzzifikasi dan menggunakan 25 rules, SBR memiliki *risetime/fall time* maksimum sebesar 1000ms dan *setting time* maksimum sebesar 9000ms dan SBR mampu mencapai kesetimbangannya kembali (*steady state*) setelah mendapatkan gangguan dari luar. Pada penelitian ini juga memiliki keterbatasan pada pembacaan sensor yang sedikit kurang akurat dikarenakan hanya menggunakan sebuah sensor yaitu *Accelerometer*.

Dengan demikian, berdasarkan tesis yang dibuat oleh Andra Laksana, kontrol robot kesetimbangan dapat menggunakan kontrol lain seperti *Fuzzy Logic*. Tetapi pada penelitian lain yang telah dibuat sebelumnya dengan menggunakan kontrol *Fuzzy Logic*, memiliki permasalahan pada kemampuan pembacaan sensor karena hanya menggunakan sensor *Accelerometer*. Jadi, berdasarkan ini hingga didesain dan dibuat *Balancing Robot Beroda Dua* menggunakan metode PI dan menggunakan sensor *accelerometer* dan *gyroscope* yang

digunakan oleh Andra Laksana untuk memperbaiki permasalahan yang terdapat pada kontrol yang dibuat oleh Handry Khoswanto [3].

### 3.1 *Inverted Pendulum*

Dasar untuk membuat robot beroda dua dapat setimbang adalah mudah yaitu dengan cara mengendalikan roda searah dengan arah jatuhnya bagian atas sebuah robot[1]. Apabila proses tersebut dapat terlaksana maka robot tersebut dapat setimbang.

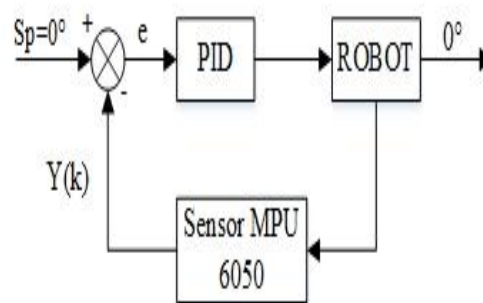


**Gambar 1. Balancing Robot Beroda Dua Menyeimbangkan Diri**

Saat balancing robot beroda dua condong kedepan 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].

### 3.2 **Kendali Proporsional Integral Derivatif (PID)**

Pengendali PID adalah suatu sistem pengendali yang merupakan gabungan antara pengendali proporsional, integral, dan turunan (derivative) [2]. Karakteristik PID *controller* sangat dipengaruhi oleh kontribusi besar dari ketiga parameter P, I dan D. Penyetelan konstanta  $K_p$ ,  $T_i$ , dan  $T_d$  akan mengakibatkan penonjolan sifat dari masing-masing elemen. Satu atau dua dari ketiga konstanta tersebut dapat diset lebih menonjol dibanding yang lain. Konstanta yang menonjol itulah akan memberikan kontribusi pengaruh pada respon sistem secara keseluruhan. Setiap kekurangan dan kelebihan dari masing-masing kontroler P, I dan D dapat saling menutupi dengan menggabungkan ketiganya secara paralel menjadi kontroler PID.



**Gambar 2. Diagram Blok Kendali PID Loop Tertutup**

Dari Gambar 2 Diagram blok pengontrolan PI loop tertutup dapat dijelaskan sebagai berikut:

1. SP = Set Point, secara sederhana maksudnya adalah suatu parameter nilai acuan atau nilai yang kita inginkan.

2.  $Y(k)$  = nilai bobot pembacaan sensor saat itu atau variabel terukur yang diumpanbalikkan oleh sensor (sinyal feedback dari sensor).
3.  $error$  = nilai kesalahan, yakni deviasi atau simpangan antar variabel terukur atau bobot sensor (PV) dengan nilai acuan (SP).
4.  $error = SP - Y(k)$

Proporsional *controler* ( $K_p$ ) akan memberikan efek mengurangi waktu naik, tetapi tidak menghapus kesalahan keadaan tunak, Integral *controler* ( $K_i$ ) akan memberikan efek menghapus keadaan tunak, tetapi berakibat memburuknya respon transien, Diferensial *controler* ( $K_d$ ) akan memberikan efek meningkatnya stabilitas sistem, mengurangi over-shoot, dan menaikkan respon transfer.

### 3.3 *Accelerometer*

*Accelerometer* berfungsi untuk mengukur percepatan, mendeteksi getaran, dan bisa juga untuk percepatan gravitasi [2]. Pendeteksian gerakan berdasarkan pada 3 sumbu yaitu kanan-kiri, atas-bawah dan depan-belakang. Pengaplikasian sensor ini biasanya untuk pengukuran kecepatan mesin, getaran mesin, getaran pada bangunan dan kecepatan yang disertai dengan pengaruh gravitasi bumi. Contoh aplikatif *accelerometer* adalah sebagai gadget elektronik, *safety installation* pada kendaraan.

Prinsip kerja *accelerometer* berdasarkan pada medan magnet yang digerakkan pada suatu konduktor ataupun konduktor yang digerakkan pada medan magnet maka akan timbul induksi elektromagnetik pada konduktor tersebut. Tipe *accelerometer* yang beredar di pasaran berbeda-beda, karena tergantung berdasarkan produsen yang memproduksi tipe *accelerometer*.

### 3.4 *Gyroscope*

*Gyroscope* berfungsi untuk mengukur atau menentukan orientasi suatu benda berdasarkan pada ketetapan momentum sudut. Dari pengertian lain *gyroscope* berfungsi untuk menentukan gerakan sesuai dengan gravitasi yang dilakukan oleh pengguna. *Gyroscope* ini memiliki peranan yang sangat penting dalam hal mempertahankan keseimbangan suatu benda seperti penggunaannya pada pesawat terbang yang dapat menentukan kemiringan pada sumbu x, y, dan z. Output yang dihasilkan oleh *gyroscope* berupa kecepatan sudut yang pada sumbu x akan menjadi phi ( $\Phi$ ), sumbu y menjadi theta ( $\theta$ ), dan sumbu z menjadi psi ( $\Psi$ ). Sebelum digunakan biasanya *gyroscope* di kalibrasi terlebih dahulu dengan menggunakan bandul yang fungsinya untuk menentukan nilai faktor ataupun dapat juga melihat pada datasheet sensor yang digunakan.

Prinsip kerja dari *gyroscope* ini adalah pada saat *gyroscope* berotasi maka *gyroscope* akan memiliki nilai keluaran. Apabila *gyroscope* berotasi searah dengan jarum jam pada sumbu Z maka tegangan output yang dihasilkan akan mengecil sedangkan jika *gyroscope* berotasi berlawanan arah dengan jarum jam pada sumbu Z maka tegangan output yang dihasilkan akan membesar. Pada saat *gyroscope* tidak sedang berotasi atau berada pada keadaan diam maka tegangan outputnya akan sesuai dengan nilai *offset gyrosensor* tersebut.

Untuk melihat data *gyroscope* dengan mikrokontroler dapat menggunakan port ADC. Nilai keluaran pada sensor diubah menjadi *radian/second* (rad/s) lalu diubah kembali menjadi *degree/second* (deg/s).

### 3.5 *Kombinasi Accelerometer dan Gyroscope*

Dengan menggunakan kombinasi *accelerometer* dan *gyroscope* pada suatu sistem maka *accelerometer* dapat memberikan pengukuran sudut saat sistem berada pada kondisi diam. Sedangkan pada saat sistem berotasi *accelerometer* tidak bisa bekerja secara maksimal karena

memiliki respon yang lambat. Kelemahan inilah yang dapat diatasi oleh *gyroscope* karena *gyroscope* dapat membaca kecepatan sudut yang dinamis. Namun *gyroscope* juga memiliki kelemahan yaitu proses perpindahan kecepatan sudut dalam jangka waktu yang panjang menjadi tidak akurat karena ada efek bias yang dihasilkan oleh *gyroscope*.

Contoh aplikatif kombinasi *accelerometer* dan *gyroscope* yaitu pada perangkat iPhone yang mengkombinasikan 2 sensor tersebut. Hal tersebut sangat membuat nyaman para pengguna dalam hal pendeteksian sensitivitas gerakan. Dari kombinasi *accelerometer* dan *gyroscope* didapatkan 6 sumbu pendeteksian yaitu 3 sumbu rotasi (x,y,z) dan 3 sumbu linier (atas-bawah, kanan-kiri, depan-belakang). Output dari kombinasi sensor ini berupa gambar yang sangat detail dan halus gerakannya dibandingkan dengan smartphone yang hanya menggunakan *accelerometer* saja.

Salah satu IC kombinasi *accelerometer* dan *gyroscope* adalah IC MPU 6050. MPU 6050 merupakan kombinasi sensor antara *accelerometer* dan *gyroscope* meskipun pada dasarnya ada sensor temperaturnya. Akses sensor ini menggunakan fitur I2c *microcontroller*. Bentuk fisik dari IC MPU 6050 dapat dilihat pada Gambar 3



Gambar 3. IC MPU 6050

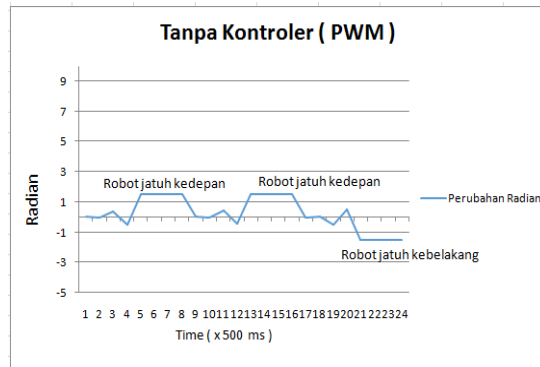
Adapun fitur-fitur MPU 6050 ini antara lain:

1. Sensitivitas *Accelerometer* yang dapat dipilih mulai 2/4/8 samapai 16 g
2. Sensitivitas *Gyroscope* yang dapat dipilih mulai 250/500/1000 sampai 2000 *degrees/s*
3. Range 16 bit untuk kedua sensor
4. Sensitivitas percepatan linier dari *Gyroscope* 0,1 derajat/s

Data rate output hingga 1000Hz, dilengkapi digital *lowpassfilter* dan memiliki frekuensi sudut maksimum 256Hz.

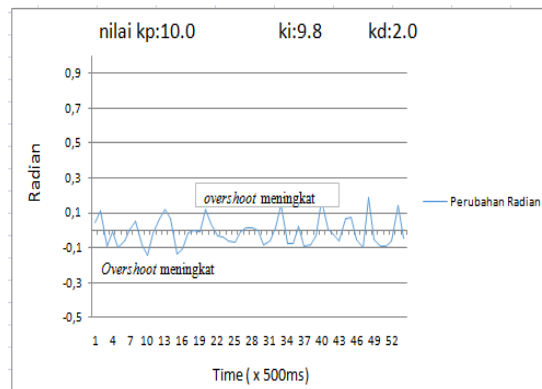
#### 4. Hasil dan Pembahasan

Pada penelitian ini dilakukan pengujian data respon robot tanpa kontroler dan menggunakan kontroler PID. Pengujian robot dengan kontroler PID dengan parameter nilai Kp diubah, nilai Ki dan Kd tetap. Pengujian robot juga dilakukan dengan parameter nilai Ki diubah, nilai Kp dan Kd tetap. Pengujian robot lainnya juga dengan parameter nilai Kd diubah, nilai Kp dan Ki tetap. Pengujian juga dilakukan dengan memberikan gangguan kepada robot.

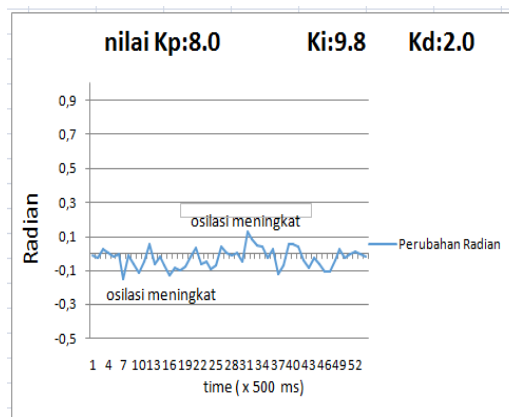


**Gambar 4. Pengujian Tanpa Kontroler**

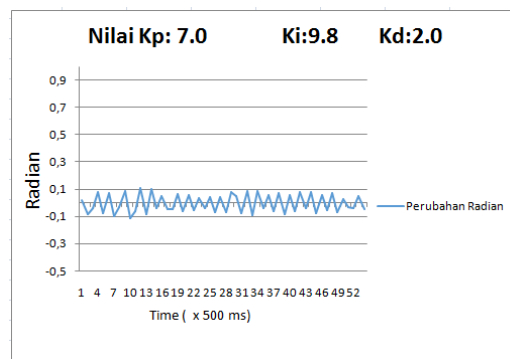
Ketika robot tidak menggunakan kontroler PID ( tanpa kontroler ) hanya menggunakan PWM, maka robot sulit mencapai keseimbangan. Hal ini dapat dilihat pada gambar 4. Pengujian mempertahankan titik keseimbangan, robot beroda dua dilakukan sebanyak 3 kali. Robot hanya mampu bertahan sekitar 2 detik, kemudian robot tidak mencapai keseimbangannya sehingga jatuh. Tanpa kontroler robot tidak dapat mencapai titik kesetimbangan  $0^\circ$ . sehingga dibutuhkan suatu kontroler yang bisa menyeimbangkan robot.



**Gambar 5. Pengujian Robot Nilai Kp10.0 Ki9.8 Kd2.0**

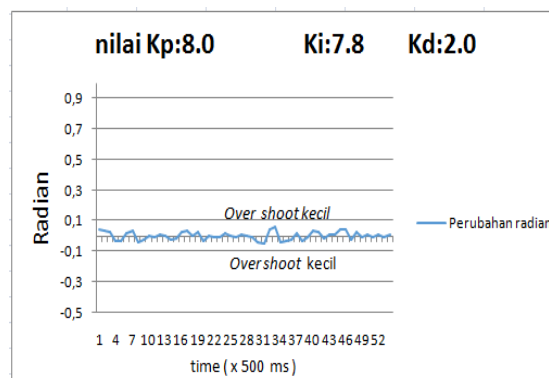


**Gambar 6. Pengujian Robot Nilai Kp8.0 Ki9.8 Kd2.0**

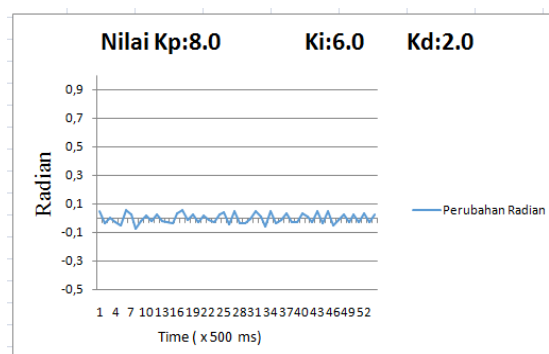


Gambar 7. Pengujian Robot Nilai Kp7.0 Ki9.8 Kd2.0

Pengujian dengan menggunakan kontroler PID dengan parameter nilai Kp diubah 10, 8 dan 7, nilai Ki tetap pada 9.8 dan nilai Kd tetap pada 2.0, yang hasilnya dapat dilihat pada gambar 5, 6 dan 7. Ketika Kp bernilai 10 *overshoot* yang terjadi meningkat dibanding Kp dengan nilai 8.0 dan 7.0. Hal ini dikarenakan nilai Mp pada Kp = 10, bernilai 18,526%. Nilai Kp paling ideal dari pengujian ini adalah 8.0.

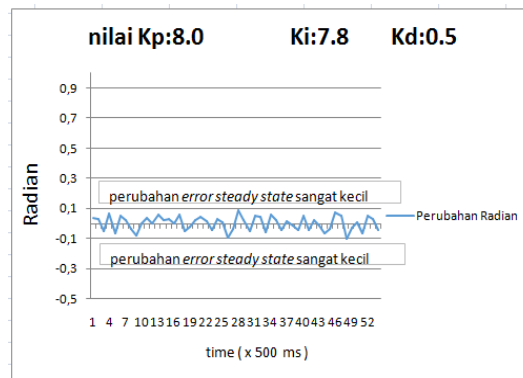


Gambar 8. Pengujian Robot Nilai Kp8.0 Ki7.8 Kd2.0



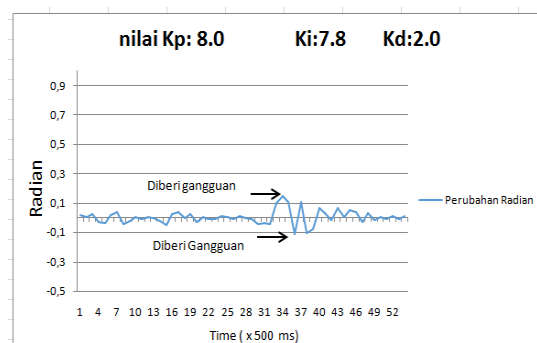
Gambar 9. Pengujian Robot nilai Kp8.0 Ki6.0 Kd2.0

Pengujian dilakukan dengan merubah parameter nilai Ki, sedangkan nilai Kp dan Kd tetap. Dari gambar 8 dan 9 dapat dilihat ketika Ki bernilai 7.8 *overshoot* yang terjadi kecil dan *error* mendekati nol. Jika dibanding dengan Ki bernilai 6.0 *error* yang terjadi masih lebih besar dan berisolasi. Nilai Ki yang paling ideal adalah 7.8.



Gambar 10. Pengujian Robot Nilai Kp8.0 Ki7.8 Kd0.5

Pengujian dilakukan dengan merubah parameter nilai Kd, sedangkan nilai Kp dan Ki tetap. Dari gambar 10 dapat dilihat bahwa *overshoot* masih ada dan *error steady state* lebih besar jika dibanding dengan gambar 8. Sehingga didapatkan nilai Ki dengan nilai paling ideal sebesar 2.0. Hasil pada Ki dengan nilai 2.0 ini nilai *overshoot* lebih kecil dan pergerakan robot dalam mencapai kesetimbangan lebih halus.



Gambar 11. Pengujian Robot Nilai Kp8.0 Ki7.8 Kd2.0 Ketika Diberi Gangguan

Pengujian dilakukan dengan memberikan gangguan ketika parameter nilai kontroler PID mencapai ideal. Gambar 11 dapat dilihat bahwa ketika diberikan gangguan yaitu pada detik ke-15 maka robot beresilasi pada detik ke-15 hingga detik ke-20. Namun setelah itu robot kembali mencapai kesetimbangan.

## 5. Kesimpulan

Setelah melakukan pengujian data dan menganalisa pengujiannya, maka dapat disimpulkan robot berhasil setimbang dengan nilai Parameter Proporsional Integral Derivatif (PID) yang paling optimal adalah Kp:8.0 Ki:7.8 Kd:2.0. Makin kecil simpangan robot artinya makin stabil robot.



**Daftar Pustaka**

- [1] Laksana, Andra. (2011). *Balancing Robot Beroda Dua Menggunakan Metode Kendali Proporsional Integral*. Tesis Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro Jl. Prof. Sudharto, Tembalang, Semarang, Indonesia
- [2] Gunawan, Ade Putra. (2012). *Robot Kesetimbangan Menggunakan Pengendali PID(Software)*. Dari Laporan TA jurusan Teknik Elektronika, Politeknik Caltex Riau, Pekanbaru, Indonesia
- [3] Utomo, Siswo Duwi. (2012). *Implementasi Metode AutoTunning PID Pada Balancing Robot*. Tesis Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro Jl. Prof. Sudharto, Tembalang, Semarang, Indonesia.
- [4] Khoswanto, Handry. (2010). *Kesetimbangan Robot Beroda Dua Menggunakan Metode Fuzzy Logic*. Laporan tesis jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Kristen Petra, Surabaya, Indonesia