

PERANCANGAN SISTEM KESEIMBANGAN *BALL AND BEAM* DENGAN MENGGUNAKAN PENGENDALI PID BERBASIS ARDUINO UNO

Else Orlanda Merti Wijaya

S1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya
e-mail : elsewijaya@mhs.unesa.ac.id

Bambang Suprianto

S1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya
e-mail : bambangsuprianto@unesa.ac.id

Abstrak

Sistem *ball and beam* merupakan salah satu contoh klasik mengenai keseimbangan, dimana sebuah bola diletakkan pada sebuah balok, bola akan sulit menuju keadaan seimbang apabila tanpa kontroler. Penelitian ini digunakan untuk mendesain sistem kendali yang dapat menyeimbangkan *ball and beam*. Sistem ini menggunakan *input* dari sensor ultrasonic. *Output* dari sensor tersebut berupa jarak atau posisi yang dikirim ke Arduino UNO. Jarak atau posisi yang didapat dibandingkan dengan nilai *setpoint* yang nilainya 0 cm. Nilai selisih dari *setpoint* dan jarak keluaran sistem dikontrol menggunakan kendali PID. Proses kendali ini diprogram pada *software* Arduino IDE yang hasilnya dikirim ke motor servo untuk mengatur kecepatan dan arah putaran motor servo untuk menyeimbangkan sistem *ball and beam*. Dari hasil pengujian diperoleh nilai parameter kontroler PID yang akan digunakan dari *tunning* nilai Kcr dengan metode Ziegler-Nichols adalah $K_p = 2.50002$, $K_i = 1.3889$ dan $K_d = 1.125$ dapat mengatasi keseimbangan pada sistem *ball and beam* mendekati nilai *setpoint*. Dengan nilai *time response* kurang dari 30 *second* saat diberikan gangguan.

Kata Kunci: *Ball And Beam*, Kontroler PID, Arduino UNO, Sensor Ultrasonik, Motor Servo

Abstract

Ball and beam system is one of the classic examples of balance, in which a ball is placed on a beam, the ball will be difficult to balance if without controller. This research is used to design control system that can keep balancing of ball and beam system. This system uses input ultrasonic sensor. The output is distance or position then sent to Arduino Uno. Distance or position comparison is given by set point has value 0 cm. Difference between setpoint and output distance of control system use PID controller. The controller run in software of Arduino IDE and the result is sent to servo motor to control speed and direction of servo motor for balancing ball and beam system. Testing is given paramater values of PID controller will be used *tunning* Kcr value with Ziegler-Nichols method is $K_p = 70$, $K_i = 466.67$ and $K_d = 2.625$ can keep balancing of ball and beam system approach setpoint. With value of time response less than 30 second when given disturbance.

Keywords: *Ball And Beam*, PID Controller, Ultrasonic Sensor, Arduino UNO, Servo Motor

PENDAHULUAN

Menjaga keseimbangan suatu obyek terhadap obyek lainnya seringkali menjadi salah satu acuan untuk menerapkan system kendali pada aspek keseimbangan. Dalam sistem kendali, pengendalian keseimbangan suatu sistem merupakan salah satu hal yang sangat penting khususnya dalam aplikasi yang berkaitan dengan gerakan seperti yang dijumpai pada aplikasi robotika (Rosalia H. Subrata, 2015). Kontrol sistem yang tidak stabil sangat penting bagi banyak masalah kontrol. System kendali keseimbangan *ball and beam*

dapat dijadikan salah satu media sederhana dalam mempelajari system kendali.

Ball and Beam merupakan salah satu contoh klasik mengenai keseimbangan, dimana sebuah bola diletakkan pada sebuah batang dan nantinya akan diberikan berbagai macam gangguan (Michael Setiawan, 2012). Pengendali PID dalam hal ini bertindak untuk menjaga posisi bola agar tetap pada posisi yang setimbang atau yang diinginkan walau diberi berbagai gangguan. Posisi atau lokasi bola akan selalu diukur dengan menggunakan sensor ultrasonik yang kemudian akan menggerakkan motor servo agar bola

berada pada titik setimbang yang diinginkan. Maka dari itu penulis memilih kontrol *Proportional Integral Derivative* (PID) sebagai kontroler yang dapat menciptakan kontrol sesuai dengan yang diharapkan penulis. Penentuan pada metode kontrol PID ini karena dengan menggunakan PID tersebut keluaran dari sistem dapat menentukan besarnya kecepatan dan arah putar motor servo yang dapat menjaga kestabilan sistem yang dibuat dalam menjaga posisi atau jarak bola yang diinginkan atau ditentukan.

Pada penelitian ini menggunakan Arduino uno dan sensor *ultrasonic* sebagai inputan yang akan mengendalikan putaran motor servo. Masalah yang didapatkan pada penelitian ini antara lain ialah merealisasikan kontrol PID pada sistem *ball and beam*, sehingga sistem ini dapat tetap mempertahankan posisi atau jarak untuk berada pada titik *set point* yang ditentukan (diinginkan) atau dengan kata lain seimbang terhadap permukaan pada bidang datar.

KAJIAN PUSTAKA

Ball and Beam

Ball and Beam merupakan salah satu contoh klasik mengenai keseimbangan, dimana sebuah bola diletakkan pada sebuah batang dan nantinya akan diberikan berbagai macam gangguan. Pengendali PID dalam hal ini bertindak sebagai pengendali gangguan sehingga posisi bola yang dari awal sudah ditentukan tidak terganggu oleh berbagai gangguan. Salah satu pengembangan yang dapat kita lihat secara nyata adalah pada bidang robotika dimana sistem *ball and beam* ini menjadi dasar kesetimbangan dari setiap mekanisme pergerakan robot. Selain robot tentu masih banyak hal yang dapat kita kembangkan dari sistem *ball and beam* ini.

Prinsip kerja *ball and beam* adalah dengan menjaga keseimbangan sistem. Untuk menjaga *ball and beam* ini seimbang atau mencapai *set point*, kontroler perlu mengetahui posisi setimbang bola atau nilai *set point* yang diinginkan, sehingga kontroler dapat memerintahkan motor servo bergerak dengan sudut tertentu yang dibutuhkan agar *beam* menuju keadaan setimbang dan posisi bola pada nilai *set point* yang diinginkan seperti yang terlihat pada Gambar 1.



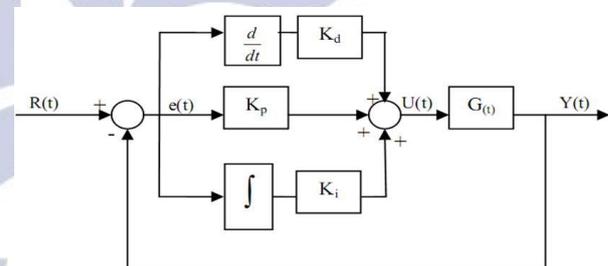
Gambar 1. Asumsi *Ball And Beam* dalam Keadaan Seimbang

Pengukuran posisi atau jarak bola akan sangat mempengaruhi arah putar motor servo dalam kecepatan tertentu. Nilai inilah yang harus dipertimbangkan sistem kontrol untuk diberikan kepada motor servo agar mengatur arah putar dan kecepatan motor servo supaya kesetimbangan (mencapai nilai *set point*) dapat terjadi.

Apabila pengukuran jarak atau posisi bola lebih dari nilai *set point* yang ditentukan maka motor servo akan berputar searah jarum jam dengan sudut tertentu agar posisi atau jarak bola menuju pada posisi setimbangnya atau nilai *set point*. Demikian sebaliknya bila kurang dari jarak atau posisi *set point* yang ditentukan, maka motor servo akan berputar berlawanan arah jarum jam secara proposional.

Sistem Pengendali PID

Pengendali PID adalah sistem pengendali gabungan dari tiga macam kontroler, yaitu K_p (*Proportional Controller*), K_i (*Integrall Controller*) dan K_d (*Derivative Controller*) seperti pada Gambar 2. Dalam pengaplikasiannya, masing-masing pengendali dapat berdiri sendiri atau dapat melakukan pengkombinasian. Dalam perancangan sistem kontrol PID yang perlu dilakukan adalah mengatur parameter P, I dan D agar respon sinyal keluaran sistem terhadap masukan tertentu sebagaimana yang diinginkan (Royyan, 2015).



Gambar 2. Kontroller PID

Berikut ini merupakan penjelasan singkat dari parameter-parameter yang ada pada kontrol PID :

a) Kontrol Proporsional

Kontrol proporsional umumnya dinyatakan dengan sebuah gain tanpa memberikan efek dinamik kepada kinerja kontroler. Adanya kontroler proporsional ini mengakibatkan sinyal *error* akan semakin besar sehingga memperbesar kemungkinan terjadinya *overshoot*. Namun kontrol proporsional dapat mengakibatkan keluaran sistem lebih cepat mencapai *setpoint*. Kontrol proporsional memperbaiki *rise time* dan *settling time* dari sebuah sistem.

b) Kontrol Integral

Kontrol integral digunakan untuk menghilangkan nilai *offset* yang biasanya dihasilkan oleh kontrol proporsional. Namun pemilihan nilai K_i yang tidak tepat dapat menyebabkan ketidakstabilan sistem. Jika pemilihan nilai K_i terlalu tinggi maka akan menyebabkan nilai keluaran akan berosilasi. Penambahan kontrol integral ini juga akan membuat respon dari sistem menjadi lambat. Oleh sebab itu maka dalam implementasinya kontrol integral dikombinasikan dengan kontrol proporsional.

c) Kontrol *Derivative*

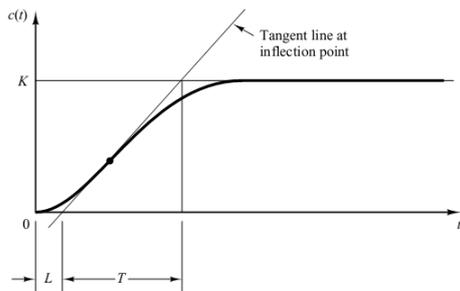
Keuntungan menggunakan kontrol *derivative* yaitu dapat merespon perubahan *error* aktuator dan dapat menghasilkan koreksi yang signifikan sebelum magnitude dari *error* aktuatornya menjadi sangat besar. Seakan-akan kontrol *derivative* ini mampu memprediksi *error* yang akan terjadi sebagai efek dari perhitungan *error* yang sebelumnya. Karena kontrol *derivative* ini bekerja berdasarkan laju perubahan *error* aktuatornya dan bukan pada *error* aktuator itu sendiri maka kontrol *derivative* ini tidak dapat berdiri sendiri.

Aturan Ziegler-Nichols Untuk Tuning PID Controllers

Metode tuning Ziegler Nichols dilakukan secara eksperimen (asumsi model belum diketahui). Berikut Aturan Ziegler-Nichols dibagi menjadi dua metode yaitu:

a) Metode Pertama Ziegler-Nichols

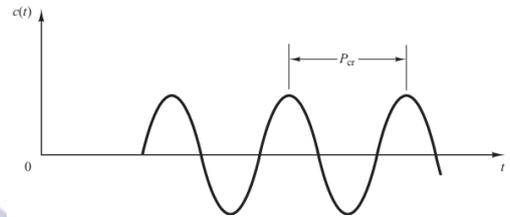
Pada metode ini, percobaan dilakukan dengan menggunakan eksperimen respon plant dari input unit step. Metode ini berlaku jika respon terhadap inputan step kurva berbentuk S melengkung seperti yang ditunjukkan dalam Gambar 3.



Gambar 3. Kurva Respon Bentuk S (Katsuhiko Ogata, 2010)

b) Metode Kedua Ziegler-Nichols

Pada metode ini, percobaan dilakukan dengan menggunakan kontroler proporsional saja. Dengan cara memberikan (*tuning*) nilai *Critical gain* (K_{cr}) yang menghasilkan grafik osilasi konsisten. Kemudian dari nilai K_{cr} ini didapatkan nilai *Critical Period* (P_{cr}) berdasarkan osilasi seperti pada Gambar 4.



Gambar 4. Osilasi Konsisten Dengan Periode P_{cr} (Katsuhiko Ogata, 2010)

Arduino UNO

Arduino UNO merupakan sebuah perangkat mikrokontroler berbasis ATmega328. Seperti halnya mikrokontroler lain, Arduino UNO juga memiliki fasilitas dasar dari mikrokontroler.



Gambar 5. Arduino UNO

Seperti yang ditunjukkan pada Gambar 5 Arduino UNO memiliki 14 pin *input/output* digital, dengan 6 diantaranya bisa digunakan sebagai PWM (*Pulse With Modulation*), 6 pin *input* analog, ICSP header, 16 MHz kristal osilator, *port* USB dan tombol reset.

Sensor Ultrasonic

Pada penelitian ini digunakan sensor ultrasonic jenis PING Parallax yang menggunakan 3 pin yaitu, transmitter gelombang ultrasonic, receiver gelombang ultrasonic, dan rangkaian control seperti pada Gambar 6.



Gambar 6. Sensor Ultrasonic PING Parallax

Sensor ini cocok untuk aplikasi elektronik yang memerlukan deteksi jarak termasuk untuk diaplikasikan untuk mengukur jarak bola di atas papan atau *beam* seperti dalam penelitian ini.

Motor Servo

Pada penelitian ini menggunakan jenis motor servo SG90. Motor servo ini memiliki spesifikasi tegangan 5 volt, agar mudah diaplikasikan bersamaan dengan Arduino yang juga memiliki spesifikasi tegangan 5 volt dan sudut putar 180 derajat, sudut ini cukup untuk mengatur kemiringan dari papan utama atau *beam* yang akan bergerak menyeimbangkan posisi bola di atasnya agar mencapai nilai *set point* yang diinginkan. Motor servo SG90 yang dipakai dalam penelitian ini ditunjukkan pada Gambar 7 sebagai berikut.



Gambar 7. Motor Servo SG90

METODE PENELITIAN

Pendekatan Penelitian

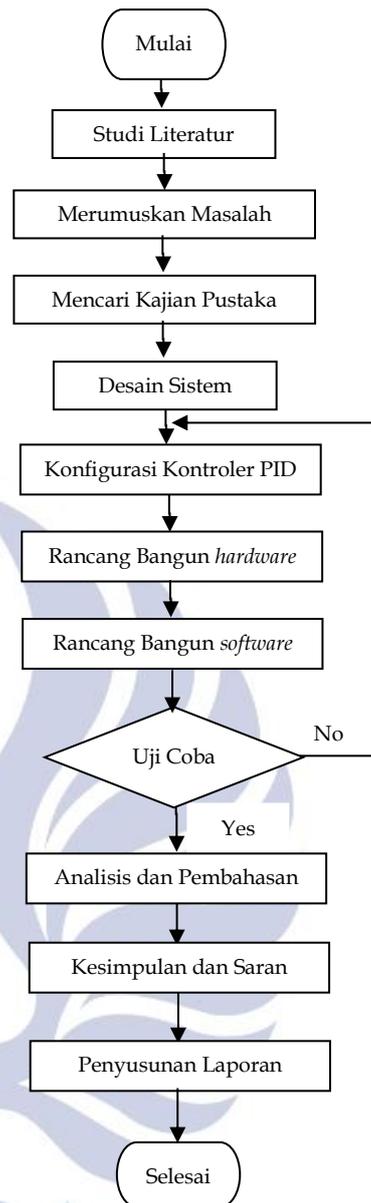
Penelitian ini untuk mengimplementasikan kontroler PID pada perancangan sistem keseimbangan *ball and beam* berbasis Arduino UNO. Pada penelitian ini, akan dilakukan uji coba kontroler PID dengan nilai parameter berbeda untuk mencari kontroler PID dengan respon paling mendekati *setpoint* dari sistem keseimbangan *ball and beam* menggunakan *software* Arduino IDE versi 1.6.12.

Tempat dan Waktu Penelitian

Penelitian dilaksanakan di Laboratorium Teknik Kendali Jurusan Teknik Elektro Universitas Negeri Surabaya dan waktu pelaksanaannya dilakukan pada semester ganjil 2017/2018.

Teknik Analisis Data

Analisis data yang diperoleh dalam penelitian ini bertujuan untuk menjawab permasalahan dalam rangka merumuskan kesimpulan, seperti dijelaskan pada diagram alir Gambar 8. sebagai berikut:



Gambar 8. Diagram Alir Langkah-Langkah Penelitian

HASIL DAN PEMBAHASAN

Metode yang digunakan dalam melakukan perancangan kontroler PID adalah menggunakan metode Ziegler-Nichols *tunning trial and error* untuk mendapatkan sebuah kontroler PID yang sesuai dalam melakukan kendali keseimbangan pada *ball and beam* agar sesuai dengan nilai *setpoint*. Kemudian dari nilai kontroler PID yang didapat akan di implementasikan pada *ball and beam*. Dan melakukan pengambilan data keluaran respon dari *ball and beam* melalui *serial monitor* pada *software* Arduino IDE versi 1.6.12.

Metode *Tunning Trial and Error*

Pada metode ini, percobaan dilakukan dengan menggunakan kontroler proporsional saja. Dengan cara memberikan (*tunning*) nilai K_p yang menghasilkan

grafik osilasi konsisten. Nilai kontroler *gain* ini disebut nilai *critical gain* (K_{cr}). Kemudian dari nilai K_{cr} ini didapatkan nilai periode kritis (P_{cr}) berdasarkan osilasi konsisten seperti pada Gambar 4.

Setelah didapatkan nilai P_{cr} maka langkah selanjutnya adalah mencari nilai K_p , K_i dan K_d dengan cara menggunakan Tabel 1.

Tabel 1. Penentuan Parameter PID

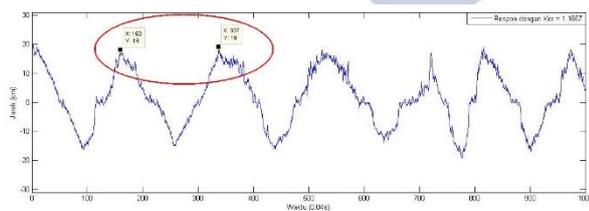
Tipe Kontroler	K_p	T_i	T_d
P	$0.5 K_{cr}$	0	0
PI	$0.45 K_{cr}$	$\frac{1}{1.2} P_{cr}$	0
PID	$0.6 K_{cr}$	$0.5 P_{cr}$	$0.125 P_{cr}$

Pengujian Sistem Keseimbangan *Ball and Beam*

Pengujian sistem *ball and beam* dilakukan dengan mengimplementasikan nilai kontroler PID berdasarkan perhitungan dari Tabel 1. yang dihasilkan dari osilasi konsisten saat melakukan *tunning* nilai K_{cr} pada sistem *ball and beam*. Untuk mencari nilai kontroler PID yang menghasilkan *error* sekecil mungkin dari nilai *setpoint* maka dilakukan *tunning* nilai K_{cr} yang berbeda untuk dilakukan perbandingan dalam mencari nilai kontroler PID yang paling kecil nilai *error*-nya dari nilai *setpoint*.

Pengujian dengan Nilai K_{cr} Pertama

Pada pengujian keseimbangan *ball and beam* ini nilai K_{cr} yang diberikan adalah 1.1667 dan menghasilkan grafik osilasi seperti pada Gambar 9.



Gambar 9. Nilai P_{cr} dari Osilasi Konsisten dengan Nilai $K_{cr}=1.1667$

Gambar 9. menunjukkan nilai P_{cr} yang didapatkan dari grafik osilasi konsisten menggunakan nilai $K_{cr} = 1.1667$. Pada grafik tersebut menunjukkan nilai X_1 dan X_2 masing-masing adalah $X_1 = 160$ dan $X_2 = 337$. Jadi nilai P_{cr} yang didapatkan dari grafik tersebut adalah selisih dari nilai X_1 dan X_2 yaitu $177 \times 0.04s = 7.08 s$.

Pada Tabel 1 disebutkan bahwa untuk mencari nilai kontroler PID ada tiga tahap yaitu mencari nilai K_p , K_i dan K_d . Untuk mencari nilai K_p , K_i dan K_d menggunakan persamaan sebagai berikut:

$$K_p = 0.6 \times K_{cr} \quad (1)$$

$$K_i = \frac{K_p}{T_i} = \frac{K_p}{0.5 \times P_{cr}} \quad (2)$$

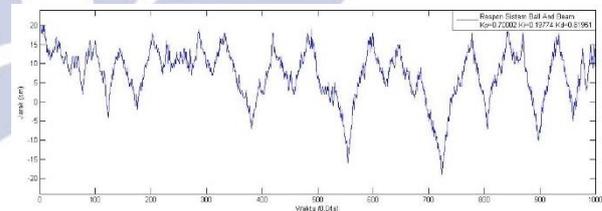
$$K_d = K_p \times T_d = K_p \times (0.125 \times P_{cr}) \quad (3)$$

Setelah mendapatkan nilai P_{cr} kemudian langkah selanjutnya adalah mencari nilai parameter PID untuk kontroler yang akan digunakan berdasarkan nilai *tunning* K_{cr} yang telah ditentukan. Persamaan yang akan digunakan adalah persamaan (1), (2) dan (3). Nilai K_{cr} yang dipakai adalah 1.1667. Sedangkan untuk nilai P_{cr} menggunakan 7.08 s berdasarkan dari hasil grafik osilasi pada Gambar 9. Dari nilai tersebut maka didapatkan nilai K_p , K_i dan K_d pada Tabel 2 berikut:

Tabel 2. Parameter PID Pertama

Nilai K_{cr}	Nilai P_{cr}	Kontroler PID
1.1667	7.08	$K_p = 0.70002$ $K_i = 0.19774$ $K_d = 0.61951$

Berdasarkan hasil dari perhitungan parameter PID menggunakan persamaan (1), (2) dan (3), maka didapatkan nilai parameter $K_p = 0.70002$, $K_i = 0.19774$ dan $K_d = 0.61951$. Kemudian di implementasikan pada sistem *ball and beam* untuk mengetahui grafik respon dari kontroler PID seperti yang bisa dilihat pada Gambar 10.



Gambar 10. Grafik Respon Sistem *Ball And Beam* dengan $K_p=0.70002$, $K_i=0.19774$, $K_d=0.61951$

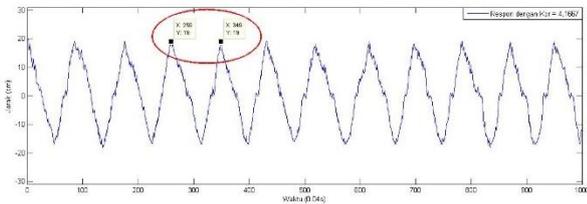
Grafik respon dari Sistem *ball and beam* ditunjukkan pada Gambar 10 adalah respon sistem *ball and beam* yang menggunakan parameter $K_p = 0.70002$, $K_i = 0.19774$ dan $K_d = 0.61951$. Untuk mengetahui nilai rata-rata dari respon sistem *ball and beam* maka dilakukan penjumlahan dari data yang didapat dibagi dengan banyaknya data adalah 7.184 cm. Maka berdasarkan hasil dari perhitungan *error* menggunakan persamaan (4) maka didapat nilai *error* sebesar 7.184. Untuk menghitung nilai *error* menggunakan persamaan sebagai berikut:

$$Error = setpoint - \text{rata-rata hasil respon} \quad (4)$$

Berdasarkan grafik yang ditunjukkan pada Gambar 10. untuk analisis respon transiennya dapat dilihat bahwa sistem mengalami osilasi yang tidak stabil dan juga beberapa *overshoot*. *Steady state error* yang dihasilkan cukup besar yaitu 7.184 cm, sedangkan nilai *setpoint* adalah 0 cm.

Pengujian dengan Nilai Kcr Kedua

Pada pengujian sistem *ball and beam* yang kedua akan dilakukan *tunning* dengan nilai Kcr yang berbeda agar didapatkan nilai kontroler PID yang berbeda pula untuk mengetahui respon nilai *error* dari sistem *ball and beam* jika diberikan nilai parameter kontroler PID yang berbeda. *Tunning* nilai Kcr yang akan diberikan adalah 4.1667 dan menghasilkan grafik respon yang akan ditunjukkan pada Gambar 11.



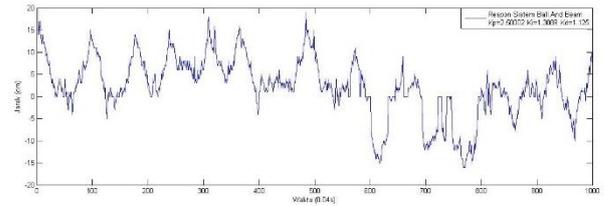
Gambar 11. Nilai Pcr dari Osilasi Konsisten dengan Nilai Kcr=4.1667

Dari grafik yang ditunjukkan pada Gambar 11. maka diketahui nilai X1 = 259 dan nilai X2 = 349, nilai Pcr adalah jarak antara periode X1 dan X2, maka didapatkan nilai Pcr adalah selisih dari nilai X1 dan X2 yaitu $90 \times 0.04s = 3.6$ s. Dari nilai Kcr dan Pcr tersebut maka didapatkan nilai Kp, Ki dan Kd dari persamaan (1), (2) dan (3) adalah sebagai berikut:

Tabel 3. Parameter PID Kedua

Nilai Kcr	Nilai Pcr	Kontroler PID
		$K_p = 2.50002$
4.1667	3.6	$K_i = 1.3889$
		$K_d = 1.125$

Jadi, nilai parameter kontroler PID yang akan digunakan pada *self balancing* robot adalah $K_p = 2.50002$, $K_i = 1.3889$, $K_d = 1.125$. Setelah didapatkan nilai parameter kontroler PID maka dilakukan pengujian sistem *ball and beam* dan pengambilan data pada serial monitor Arduino IDE versi 1.6.12. Kemudian data yang didapat akan di *plot* agar mendapatkan grafik hasil respon sistem *ball and beam* kemudian dilakukan analisis antara *setpoint* dan *error* yang dihasilkan dari pengujian sistem *ball and beam*.



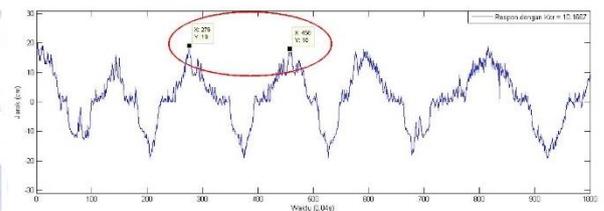
Gambar 12. Grafik Respon Sistem *Ball And Beam* dengan $K_p=2.50002$, $K_i=1.3889$, $K_d=1.125$

Gambar 12. menunjukkan grafik respon dari sistem *ball and beam* dengan menggunakan nilai $K_p = 2.50002$, $K_i = 1.3889$ dan $K_d = 1.125$. Berdasarkan hasil respon yang dihasilkan dari pengumpulan data yang diambil dari serial monitor *software* Arduino IDE, maka didapatkan hasil rata-rata dari jumlah data dibagi dengan banyaknya data adalah 2.242 cm. Maka berdasarkan hasil dari perhitungan *error* menggunakan persamaan (4) maka didapat nilai *error* sebesar 2.242.

Berdasarkan grafik pada Gambar 12. untuk analisis respon transiennya dapat dilihat bahwa sistem mengalami osilasi yang tidak stabil dan beberapa *overshoot*. *Steady state error* yang dihasilkan juga cukup kecil yaitu kurang dari 2.5 cm yaitu 2.242 cm, sedangkan nilai *setpoint* adalah 0 cm.

Pengujian dengan Nilai Kcr Ketiga

Pada tahap pengujian sistem *ball and beam* yang ketiga ini akan menggunakan *tunning* nilai Kcr=10.1667 untuk mendapatkan nilai parameter PID yang berbeda agar dapat dianalisis hasilnya untuk mencari kontrol PID yang sesuai dengan sistem *ball and beam* yang dibuat. Grafik respon sistem *ball and beam* menggunakan Kcr=10.1667 ditunjukkan pada Gambar 13.



Gambar 13. Nilai Pcr dari Osilasi Konsisten dengan Nilai Kcr=10.1667

Gambar 13. menunjukkan nilai Pcr yang didapatkan dari grafik osilasi konsisten menggunakan nilai Kcr = 10.1667. Pada grafik tersebut menunjukkan nilai X1 dan X2 masing-masing adalah X1 = 276 dan X2 = 458. Jadi nilai Pcr yang didapatkan adalah selisih dari nilai X1 dan X2 yaitu $182 \times 0.04s = 7.28$ s.

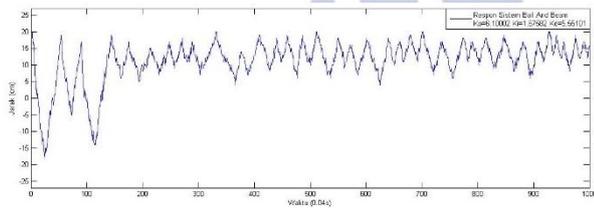
Setelah mendapatkan nilai Pcr langkah selanjutnya adalah mencari nilai parameter PID untuk kontroler yang akan digunakan dengan menggunakan persamaan

(1), (2) dan (3). Nilai Kcr yang dipakai adalah 10.1667 dan nilai Pcr menggunakan 7.28 s. Dari nilai tersebut maka didapatkan nilai Kp, Ki dan Kd sebagai berikut:

Tabel 4. Parameter PID Ketiga

Nilai Kcr	Nilai Pcr	Kontroler PID
10.1667	7.28	Kp = 6.10002 Ki = 1.67582 Kd = 5.55101

Berdasarkan hasil dari perhitungan parameter PID menggunakan persamaan (1), (2) dan (3), maka didapatkan nilai parameter Kp = 6.10002, Ki = 1.67582 dan Kd = 5.55101. Dari nilai parameter PID tersebut kemudian di implementasikan pada sistem *ball and beam* dan melakukan pengambilan data untuk mengetahui grafik respon dari kontroler PID seperti yang bisa dilihat pada Gambar 14.



Gambar 14. Grafik Respon Sistem *Ball And Beam* dengan Kp=6.10002, Ki=1.67582, Kd=5.55101

Gambar 14. menunjukkan grafik respon dari sistem *ball and beam* dengan menggunakan nilai Kp = 6.10002, Ki = 1.67582 dan Kd = 5.55101 dengan nilai *setpoint* 0 cm. Berdasarkan hasil respon yang dihasilkan dari data yang diambil dari serial monitor *software* Arduino IDE, maka didapatkan hasil rata-rata dari jumlah data dibagi dengan banyaknya data adalah 11.273 cm. Maka berdasarkan hasil dari perhitungan *error* menggunakan persamaan (4) maka didapat nilai *error* sebesar 11.273.

Berdasarkan grafik pada Gambar 14, untuk analisis respon transiennya dapat dilihat bahwa sistem mengalami osilasi yang sedikit dan tidak stabil, kemudian masih banyak terjadi *overshoot*. *Steady state error* yang dihasilkan juga cukup besar yaitu lebih dari 10 cm, sedangkan nilai *setpoint* adalah 0 cm.

Sehingga dari semua respon yang ditunjukkan sistem *ball and beam* dengan menggunakan nilai Kcr yang berbeda agar mendapatkan parameter PID yang berbeda pula dapat disimpulkan berdasarkan Tabel 5. di bawah ini:

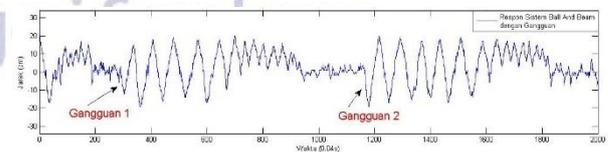
Tabel 5. Respon Sistem *Ball And Beam* Menggunakan Nilai Kcr yang Berbeda

No	Nilai Kcr	Parameter PID	Error (derajat)
1.	Kcr = 1.1667	Kp = 0.70002 Ki = 0.19774 Kd = 0.61951	7.184
2.	Kcr = 4.1667	Kp = 2.50002 Ki = 1.3889 Kd = 1.125	2.242
3.	Kcr = 10.1667	Kp = 6.10002 Ki = 1.67582 Kd = 5.55101	11.273

Berdasarkan hasil respon sistem *ball and beam* yang ditunjukkan pada Tabel 5. yang memiliki nilai *error* paling kecil adalah respon dari *tunning* dengan nilai Kcr=4.1667 yaitu memiliki nilai *error* 2.242 cm paling mendekati dengan nilai *setpoint* yaitu 0 cm. Jadi, parameter PID yang akan digunakan dalam sistem *ball and beam* yang telah dibuat adalah kontroler dengan nilai Kp = 2.50002, Ki = 1.3889 dan Kd = 1.125.

Pengujian Sistem *Ball And Beam* dengan Gangguan

Setelah melakukan *tunning* parameter PID maka didapatkan kontroler PID dengan respon yang cukup baik dengan memiliki nilai *error* yang kecil. Kemudian kontroler PID tersebut di implementasikan pada sistem *ball and beam* dan dilakukan pengujian untuk mengetahui bahwa kontroler tersebut mampu mengatasi gangguan dari keseimbangan sistem agar tetap berada pada titik *set point* yang ditentukan. Gangguan yang dimaksudkan disini adalah ketika bola sudah berada di titik seimbang maka diberikan sedikit dorongan kepada bola dan apakah bola masih dapat kembali pada posisi seimbang akan ditunjukkan dari grafik respon pada Gambar 15. berikut ini:

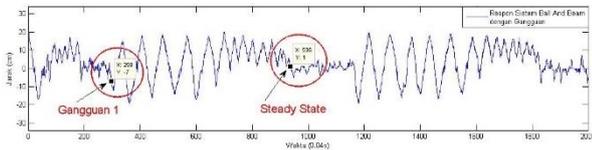


Gambar 15. Respon Sistem *Ball And Beam* dengan Diberikan Gangguan

Gambar 15. menunjukkan respon dari sistem *ball and beam* ketika diberikan gangguan. Pada pengujian sistem *ball and beam* ini menggunakan parameter kontroler PID dengan nilai Kp = 2.50002, Ki = 1.3889 dan Kd = 1.125 yang telah ditentukan berdasarkan *tunning* nilai Kcr. Ketika sistem *ball and beam* diberikan gangguan, keseimbangan sistem dapat diatasi oleh

kontroler PID dengan cara menggerakkan motor servo untuk menyeimbangkan atau membuat posisi bola berada pada jarak yang diinginkan (*set point*) kembali.

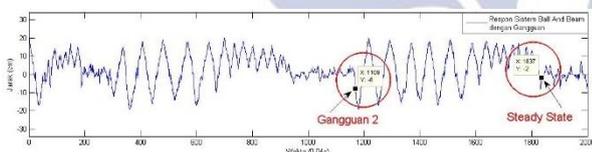
Untuk analisis respon sistem *ball and beam* terhadap gangguan yang diberikan maka dapat ditunjukkan dengan Gambar 16. berikut ini:



Gambar 16. *Time Response* Gangguan Pertama

Gambar 16. menunjukkan *time response* menuju *steady state* dari sistem *ball and beam* ketika diberikan gangguan pertama. Dari respon tersebut maka diketahui bahwa awal waktu sistem *ball and beam* diberikan gangguan sampai respon *steady state* adalah $X1 = 298$ menunjukkan waktu awal diberikan gangguan, $X2 = 936$ menunjukkan respon *steady state*. Maka *Time Response* atau waktu yang dibutuhkan sistem *ball and beam* untuk mengatasi gangguan yang diberikan ketika dalam keadaan seimbang sampai menuju *steady state* adalah selisih dari nilai $X1$ dan $X2$ yaitu $638 \times 0.04s = 25.52$ s.

Kemudian untuk analisis *time response* gangguan kedua pada grafik yang ditunjukkan pada Gambar 15. dapat ditunjukkan pada Gambar 17. berikut ini:



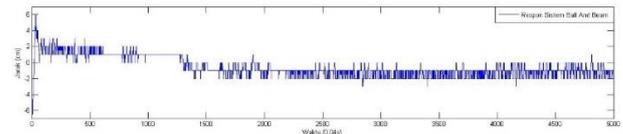
Gambar 17. *Time Response* Gangguan Kedua

Berdasarkan grafik yang ditunjukkan pada Gambar 17. maka diketahui bahwa awal waktu sistem *ball and beam* diberikan gangguan sampai respon *steady state* adalah $X1 = 1169$ menunjukkan waktu awal diberikan gangguan, $X2 = 1837$ menunjukkan respon *steady state*. Maka *Time Response* atau waktu yang dibutuhkan sistem *ball and beam* untuk mengatasi gangguan yang diberikan ketika dalam keadaan seimbang sampai menuju *steady state* adalah selisih dari nilai $X1$ dan $X2$ yaitu $668 \times 0.04s = 26.72$ s. Menunjukkan bahwa respon dari sistem *ball and beam* sudah sesuai dari yang diharapkan dan mampu mengatasi gangguan yang diberikan karena pada saat sistem *ball and beam* diberikan gangguan sistem masih bisa mengatasi gangguan tersebut dengan parameter kontroler PID yang telah ditentukan sebelumnya. Dengan nilai *error* yang kecil yaitu 3.1855 cm dan *time response* menuju *steady state* kurang dari 30 s yang berarti sistem *ball and beam*

masih dapat berada pada posisi atau jarak seimbang meskipun sudah diberikan gangguan dengan adanya parameter kontroler PID yang telah dibuat.

Pengujian Sistem *Ball And Beam* tanpa Gangguan

Setelah melakukan pengujian sistem *Ball And Beam* dengan penambahan gangguan berupa dorongan sedikit terhadap bola maka disini juga dilakukan pengujian sistem *Ball And Beam* dengan tanpa adanya penambahan gangguan untuk mengetahui dari kestabilan sistem ini.



Gambar 18. Respon Sistem *Ball And Beam* Tanpa Gangguan

Pada Gambar 18. di atas menunjukkan respon dari sistem *ball and beam* yang dilakukan ketika dalam keadaan seimbang pada rentan waktu 200 *second* tanpa adanya penambahan gangguan, dimana dari hasil respon yang didapat sudah sesuai dengan yang diharapkan untuk mampu mempertahankan kondisi jarak atau posisi bola untuk tetap berada dalam keadaan seimbang atau mendekati nilai *set point* yang diinginkan.

PENUTUP

Simulan

Dari hasil pengujian yang dilakukan untuk perancangan sistem keseimbangan *ball and beam* dengan menggunakan pengendali PID berbasis Arduino UNO menggunakan *tunning trial and error* nilai Kcr metode Ziegler-Nichols dengan nilai konstanta $Kp = 2.50002$, $Ki = 1.3889$ dan $Kd = 1.125$ bola dapat berada pada posisi atau jarak seimbang yang diinginkan dengan nilai *error* yang dihasilkan kecil yaitu 2.242 cm dari *tunning* nilai $Kcr = 4.1667$ sedangkan untuk *time response* menuju *steady state* juga baik yaitu kurang dari 30 *second* untuk itu sistem dengan metode ini dapat dikatakan berhasil

Saran

Dapat dikembangkan dengan mengubah jenis metode pengontrolan yang lain, seperti adaptif, fuzzy logic, jaringan saraf tiruan (JST), dll agar dapat mengurangi nilai *error steady state*.

Penambahan sensor atau dengan penggantian sensor jenis lain yang lebih akurat agar pembacaan posisi bola lebih presisi.

DAFTAR PUSTAKA

- Ogata, Katsuhito. (2010). *Modern Control Engineering RD 5 Edition*. New Jersey. Prentice Hall.
- Royyan, M., dkk. 2015. “*Implementation of Kalman Filter and PID Controller for Inverted Pendulum Robot*”.
https://openlibrary.telkomuniversity.ac.id/pustaka/files/100391/jurnal_eproc/implementasi-kalman-filter-dan-pid-kontroler-pada-robot-pendulum-terbalik.pdf. Bandung: Telkom University.
- Setiawan, Michael. 2012. *Perancangan Pengendali PID untuk Sistem Ball And Beam*. Jakarta: Universitas Trisakti.
- Subrata, Rosalia H dan Richard R. Ligianto. 2015. *Pengendalian Keseimbangan Ball and Plate Menggunakan Pengendali PID Dan Pengolahan Citra Digital*. Jakarta: Universitas Kristen Maranatha.

