

# KENDALI *PROPORTIONAL INTEGRAL DERIVATIVE* PADA KESEIMBANGAN BOLA DENGAN MENGGUNAKAN *LABVIEW* DAN *MYRIO*

Oleh: Kusno Utomo<sup>1</sup>, Afiv Setyaji<sup>2</sup>, Alyaa Qonita Arifah<sup>3</sup>  
<sup>1234</sup>Jurusan Teknik Elektro, Politeknik Negeri Semarang  
 E-mail: kusno.utomo@polines.ac.id

## Abstrak

*kendali keseimbangan bola dibuat untuk menguji keseimbangan bola diatas sebuah papan dengan menggunakan MyRio dengan kendali berbasis Proporsional Integral Derivatif (PID). Alat ini dibuat dengan menggunakan sensor infrared GP2Y0A21YK sebagai pendeteksi jarak, limit switch sebagai pendeteksi keadaan papan, dan rotary encoder untuk mendeteksi sudut kemiringan papan yang diprogram menggunakan software LabVIEW. Pemilihan parameter awal PID menggunakan manual tuning dengan menentukan nilai KP, kemudian KD dan KI. Hasil PID selanjutnya akan menggerakkan motor DC. Untuk menghilangkan noise pada pembacaan sensor infrared maka digunakan filter coil. Kinerja dari kendali PID ini didasarkan pada steady state error. Hasil dari pengujian menunjukkan rata rata error pada masing masing setpoint yaitu pada setpoint 33 cm sebesar 3%, pada setpoint 27 cm sebesar 4,9% dan pada setpoint 23 cm sebesar 4%.*

**Kata kunci :** Bola dan Papan, Infrared GP2Y0A21YK, LabVIEW, MyRio, PID.

## Abstract

*Ball balance control are made to test the balance of the ball on a board using MyRio with Proportional Integral Derivative (PID) based control. This tool made using a GP2Y0A21YK infrared sensor as a distance detector, a limit switch as a board state detector, and a rotary encoder to detect the board's tilt angle programmed using LabVIEW software. Selection of the initial PID parameters using manual tuning by determining the value of KP, then KD and then KI. The results of the PID will then drive the DC motor. To eliminate noise on infrared sensor readings, a filter coil is used. The performance of this PID control is based on steady state error. The results of the test show the average of error for each set point that is for setpoint 33cm has 3% of error, for setpoint 27 cm has 4, 9% of error and for 23 cm has 4% of error.*

**Keywords :** Ball and Beam, Infrared GP2Y0A21YK, LabVIEW, MyRio, PID

## 1. Pendahuluan

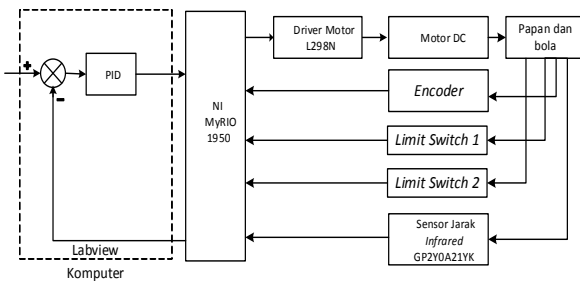
Sistem yang menjadi dasar keseimbangan pada mekanisme pergerakan robot adalah sistem bola dan papan. *Ball and Beam* merupakan salah satu contoh klasik mengenai keseimbangan, dimana sebuah bola diletakkan pada sebuah batang dan nantinya akan diberikan berbagai macam gangguan. Salah satu pengembangan yang dapat dilihat secara nyata adalah pada bidang robotika di mana sistem *ball and beam* menjadi dasar kesetimbangan dari setiap mekanisme pergerakan robot. Sistem *ball and beam* merupakan sistem yang umum digunakan sebagai alat peraga dalam pembelajaran sistem kendali. Sistem *ball and beam* bekerja dengan membaca posisi bola untuk

mengendalikan bidang persegi panjang agar bola dapat mencapai keadaan yang seimbang atau diam pada posisi yang ditentukan. Pengendalian menggunakan PID terdiri dari tiga komponen yaitu Proporsional (P), Integral (I), dan Derivative(D). Pembelajaran sistem kontrol PID pada mata kuliah Sistem Kontrol di perguruan tinggi masih menggunakan sistem manual, metode pembelajaran semacam ini memerlukan waktu yang lama dan cukup sulit dipahami oleh mahasiswa. Untuk meningkatkan pemahaman mahasiswa dalam materi sistem kontrol dibuatlah alat peraga Kendali *Proportional Integral Derivative* Pada Keseimbangan Bola Dengan Menggunakan *LabVIEW* dan *MyRio*.

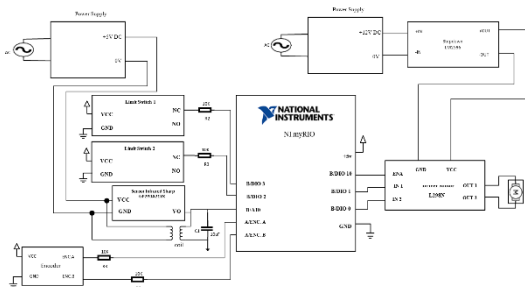
## 2. Metode Penelitian

Penelitian menggunakan metode *waterfall* yang terdiri dari analisa kebutuhan sistem, perancangan dan pembuatan perangkat keras dan perangkat lunak, uji coba sistem.

Desain kontrol pada penelitian ini menggunakan MyRio sebagai system akuisisi data dan LabVIEW sebagai perangkat lunak pemrograman kendali PID, membaca input dan menggerakkan output serta menampilkan data hasil berupa grafik pada layar komputer. Berikut gambar diagram blok sistem kendali, rangkaian keseluruhan, diagram alir program dan Tampilan Front Panel LabVIEW.



Gambar 1. Diagram Blok Sistem Kendali

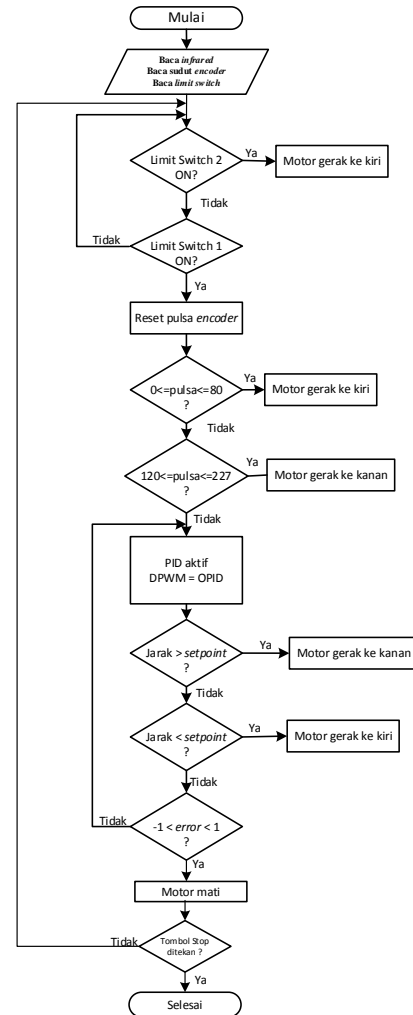


Gambar 2. Rangkaian Keseluruhan

Berikut penjelasan per blok:

- Pengaturan setpoint dan program kendali menggunakan komputer dengan software LabVIEW.
- MyRio sebagai sistem akuisisi data untuk membaca masukan dan mengaktifkan keluaran.
- Driver motor untuk mengontrol kecepatan serta arah putaran motor DC
- Motor DC sebagai penggerak papan agar bola dapat berada pada posisi seimbang yang telah ditetapkan.

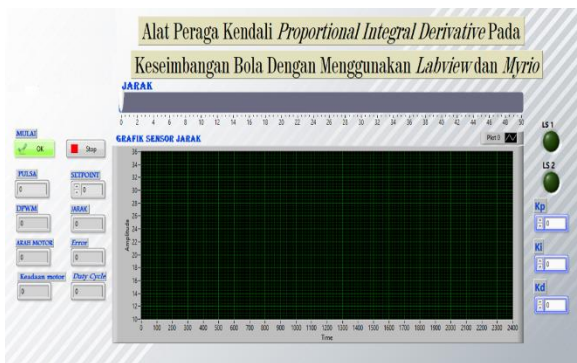
- Encoder sebagai penghasil pulsa untuk menentukan kemiringan papan.
- Sensor jarak *infrared* GP2Y0A21YK mendeteksi jarak bola diatas papan.
- Limit switch sebagai masukan untuk menggerakkan papan dan untuk membatasi pergerakan papan.



Gambar 3. Diagram Alir Alat

Pengoperasian alat dimulai dengan menekan tombol mulai pada *front panel* LabVIEW dan memasukkan nilai *setpoint*. Sebelum PID dijalankan, diperlukan batas posisi papan yang akan dibaca oleh pulsa *encoder* agar papan berada pada posisi yang sesuai untuk menjalankan PID. Jika papan berada pada pulsa diantara 0 sampai 80 (kurang dari pulsa aktif PID) maka motor DC akan menggerakkan papan ke kiri. Jika papan berada pada pulsa diantara 120 sampai 227 maka motor DC akan

menggerakkan papan ke kanan. Ketika papan berada pada pulsa diantara 80 sampai 120 maka PID akan berjalan dan motor DC akan bergerak sesuai dengan PWM hasil dari keluaran PID. Jika *error* yang terdeteksi dari sensor jarak bernilai negatif maka papan akan bergerak ke kanan. Jika *error* yang terdeteksi bernilai positif maka papan akan bergerak ke kiri. Dan jika nilai *error* antara minus satu hingga satu, jarak yang dideteksi sama dengan setpoint maka motor akan berhenti. Sistem akan berhenti ketika tombol *stop* ditekan pada *front panel LabVIEW*.

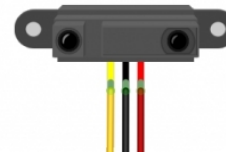


Gambar 4. Tampilan Front Panel *LabVIEW*

### 3. Kajian Pustaka

Sensor Jarak *Infrared* SHARP GP2Y0A21YK Sensor jarak Sharp pilihan populer untuk banyak proyek yang membutuhkan pengukuran jarak akurat. Sensor *infrared* ini lebih ekonomis daripada pengukur jarak sonar, juga memberikan kinerja yang jauh lebih baik daripada alternatif *infrared* lain. *Interfacing* ke mikrokontroler yang paling mudah adalah *output* analog tunggal dapat dihubungkan ke sebuah konverter analog ke digital untuk melakukan pengukuran jarak, atau *output* dapat dihubungkan ke komparator untuk deteksi ambang batas. Jangkauan deteksi versi ini adalah sekitar 10 cm sampai 80 cm. Sensor ini termasuk pada sensor jarak kategori optik. Pada dasarnya, sensor ini sama seperti sensor *infrared* konvensional, GP2Y0A21YK memiliki bagian *transmitter/emitter* dan *receiver* (detektor). Bagian *transmitter* akan memancarkan sinyal *infrared*, sedangkan pantulan dari *infrared* (apabila mengenai sebuah objek)

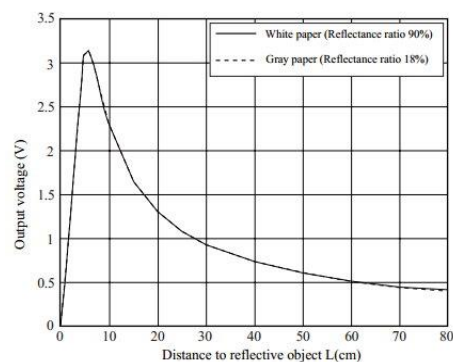
akan ditangkap oleh bagian detektor yang terdiri dari lensa pemfokus dan sebuah *linear charge-couple device* (CCD) array. *Linear* CCD array terdiri atas sederetan elemen peka cahaya yang disebut piksel (*Picture element*).[8] Dalam Gambar 11 memperlihatkan bentuk fisik dari sensor sharp GP2Y0A21YK.



Gambar 4. Sensor Jarak Infrared GP2Y0A21YK

Sensor ini memiliki 3-pin, *Voltage*, *Ground*, *Signal*. *Output* sensor ini adalah analog tunggal, dapat terhubung ke sebuah konverter analog ke digital untuk mengambil pengukuran jarak, atau *output* dapat dihubungkan ke *comparator* untuk deteksi ambang batas. Untuk menghubungkan sensor ke mikrokontroler, sensor Sharp GP2Y0A21YK menggunakan konektor JST 3 pin yang terhubung ke kabel 3 in JST untuk sensor jarak itu sendiri.[8]

Hasil *output* tegangan sensor *infrared* tidaklah linier melainkan membentuk kurva seperti pada gambar berikut. Sensor mulai menampilkan jarak yang valid saat berada di jarak sekitar 10 cm hingga 80 cm. Gambar 12 menunjukkan grafik tegangan dan jarak pada sensor *infrared* GP2Y0A21YK.



Gambar 5. Grafik Tegangan dan Jarak Sensor Infrared GP2Y0A21YK

Tegangan	Jarak Aktual	Jarak dengan rumus	Error
2,1	10	10,0787	0,780805
1,94	11	11,17654	1,579568
1,81	12	12,24604	2,009156
1,75	13	12,84023	-1,24426
1,64	14	14,16051	1,133527
1,62	15	14,43848	-3,88906
1,55	16	15,51804	-3,10583
1,48	17	16,78198	-1,29916
1,44	18	17,59632	-2,29413
1,4	19	18,48328	-2,79561
1,34	20	19,95976	-0,20162
1,3	21	21,04802	0,228121
1,26	22	22,22469	1,010993
1,23	23	23,16794	0,724864
1,19	24	24,5102	2,081596
1,17	25	25,21887	0,867869
1,13	26	26,71407	2,673023
1,09	27	28,31688	4,650518
1,07	28	29,16005	3,978207
1,05	29	30,03181	3,43571

Untuk mendapatkan hasil pembacaan sensor yang tepat maka dilakukan pengukuran tegangan sensor *infrared* terhadap jarak yang terdeteksi. Tabel 1 menunjukkan hasil pembacaan tegangan sensor *infrared*.

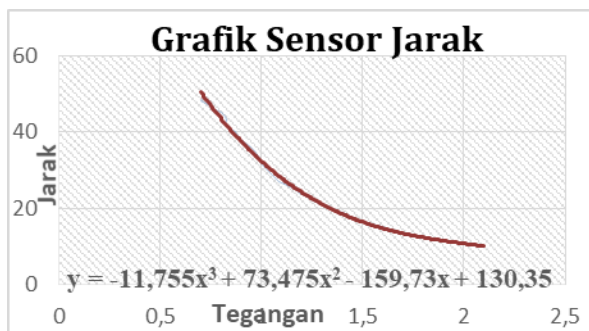
Tabel 1 Pengukuran Infrared

Tegangan	Jarak Aktual	Jarak dengan rumus	Error
1,03	30	30,93272	3,015323
1,01	31	31,86336	2,709568
1	32	32,34	1,05133
0,99	33	32,82428	-0,53533
0,98	34	33,31628	-2,05222
0,97	35	33,81606	-3,50113
0,95	36	34,83924	-3,33175
0,92	37	36,43416	-1,55304
0,9	38	37,53836	-1,2298
0,87	40	39,25742	-1,89155
0,84	41	41,05352	0,130377
0,82	42	42,29466	0,696682
0,83	43	41,66967	-3,19256
0,82	44	42,29466	-4,03205
0,8	45	43,57144	-3,27866
0,76	46	46,2342	0,506545
0,74	47	47,6213	1,304672
0,7	48	50,50979	4,968908
0,72	49	49,04631	0,09442

Berdasarkan data yang jarak terbaca dan tegangan, maka didapatkan grafik *non-linear*. Untuk mendapatkan pembacaan yang akurat maka dibutuhkan rumus polinomial dari grafik tersebut yang didapat dengan *excel*. Dari pengukuran diatas didapatkan rumus :

$$\text{Jarak} = -11,755(\text{tegangan})^3 + 73,475(\text{tegangan})^2 - 159,73(\text{tegangan}) + 130,35$$

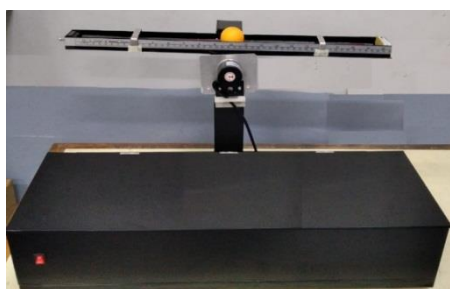
Gambar grafik *tranddline* dari pembacaan tegangan sensor jarak ditunjukkan pada Gambar 13.



Gambar 6 Grafik Sensor Jarak Infrared

### 3. Hasil dan Pembahasan

Pada percobaan kali ini, metode yang digunakan untuk kontrol PID adalah dengan *manual tuning*. Metode *manual tuning* dilakukan secara bertahap dimulai dengan menentukan nilai  $K_p$ , kemudian  $K_d$  dan  $K_i$  secara manual. Pada percobaan kali ini dilakukan percobaan tiga kali *setpoint* yang berbeda, yaitu 33 cm, 27 cm, dan 23 cm.



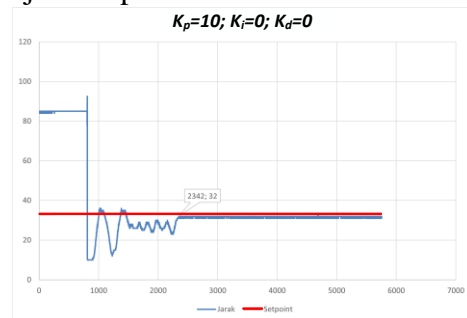
Gambar 5. Alat Peraga Kendali Keseimbangan Bola

### 3.1 Hasil

#### 3.1.1 Hasil Grafik *Setpoint* 33

Pada percobaan kali ini, *setpoint* yang ditentukan adalah 33 cm. Pada *setpoint* ini dilakukan beberapa percobaan nilai  $K_p$ ,  $K_i$ , dan  $K_d$  yang berbeda.

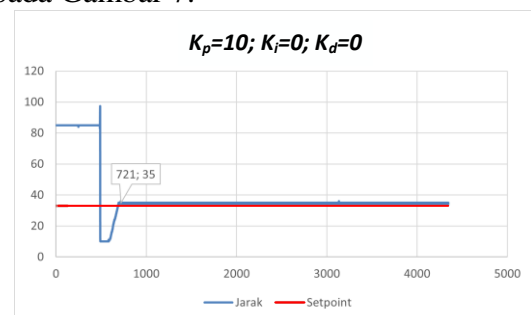
Percobaan 1 : nilai pertama yang diatur adalah nilai  $K_p$  yaitu sebesar 11. Namun terjadi osilasi yang terlalu besar sehingga nilai  $K_p$  diturunkan menjadi 10. Ketika nilai  $K_p = 10$ ,  $K_i = 0$ , dan  $K_d = 0$  papan berhenti bergerak dan seimbang dengan jarak yang terbaca sebesar 32cm dan settling time sebesar 23,42s. Respon PID ini memiliki steady state error sebesar 3,03%. Hasil grafik percobaan pertama ditunjukkan pada Gambar 6.



Gambar 6. Grafik Percobaan 1 Setpoint 33

Percobaan 2 : berdasarkan dari percobaan pertama maka saat pertama kali program berjalan nilai  $K_p$  diberikan sebesar 10. Ketika nilai  $K_p = 10$ ,  $K_i = 0$ , dan  $K_d = 0$  papan berhenti bergerak dan seimbang dengan jarak yang terbaca sebesar 35cm. Pada percobaan ini memiliki *settling time* sebesar 7,21s. Respon PID ini memiliki *steady state error* sebesar -6,06%.

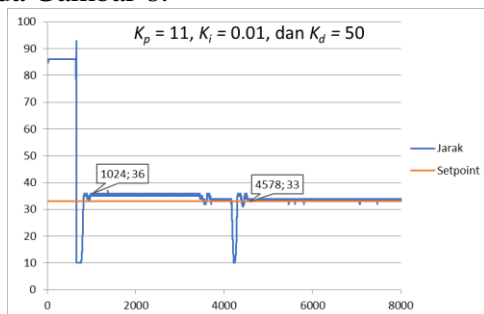
Hasil grafik percobaan kedua ditunjukkan pada Gambar 7.



Gambar 7. Grafik Percobaan 2 Setpoint 33

Percobaan 3 : nilai pertama yang diatur adalah nilai  $K_p$  yaitu sebesar 11. Namun hasilnya melebihi *setpoint*, sehingga ditambahkan nilai  $K_d = 50$  untuk mengurangi steady state error. Kemudian ditambahkan lagi nilai  $K_i = 0,01$ . Sehingga dengan nilai  $K_p = 11$ ,  $K_i = 0,01$ , dan  $K_d = 50$ , papan dalam keadaan seimbang dan bola tidak bergerak lagi. Respon PID ini memiliki *steady state error* sebesar 0%. Pada *setpoint* 33 cm, untuk mencapai keadaan bola seimbang diatas papan atau *settling time* yaitu sebesar 45,78s.

Hasil grafik percobaan ketiga ditunjukkan pada Gambar 8.



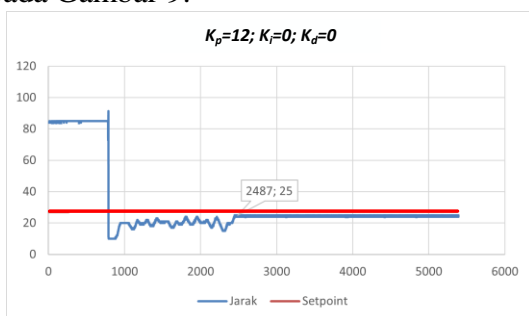
Gambar 8. Grafik Percobaan 3 Setpoint 33

### 3.1.2 Hasil Grafik Setpoint 27

Pada percobaan kali ini, *setpoint* yang ditentukan adalah 27 cm. Pada *setpoint* ini dilakukan beberapa percobaan nilai  $K_p$ ,  $K_i$ , dan  $K_d$  yang berbeda.

Percobaan 1 : nilai pertama yang diatur adalah nilai  $K_p$  yaitu sebesar 12. Dengan nilai  $K_p=12$ ,  $K_i=0$ , dan  $K_d=0$ , papan berada dalam kondisi seimbang dan bola berhenti bergerak. *Steady state error* sebesar 7,4% dan *settling time* sebesar 24,87s.

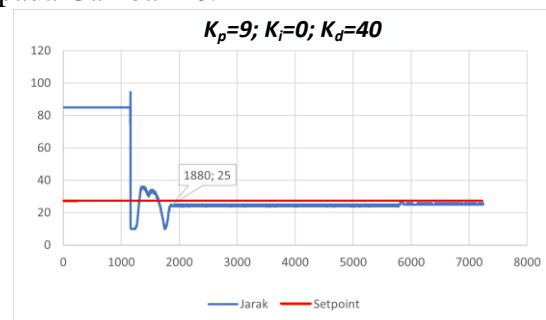
Hasil grafik percobaan pertama ditunjukkan pada Gambar 9.



Gambar 9. Grafik Percobaan 1 Setpoint 27

Percobaan 2 : nilai pertama yang diatur adalah nilai  $K_p$  yaitu sebesar 9. Kemudian ditambahkan nilai  $K_d=40$  untuk mengurangi osilasi. Dengan nilai  $K_p=9$ ,  $K_i = 0$ , dan  $K_d = 40$  papan berhenti bergerak dan seimbang dengan jarak yang terbaca sebesar 25cm dan *settling time* sebesar 18,8s. Respon PID ini memiliki *steady state error* sebesar 7,4%.

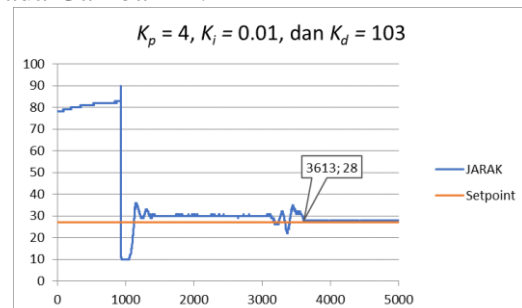
Hasil grafik percobaan kedua ditunjukkan pada Gambar 10.



Gambar 10. Grafik Percobaan 2 Setpoint 27

Percobaan 3: nilai  $K_p$  yaitu sebesar 4. Kemudian untuk mengurangi nilai *steady state error* maka ditambah nilai  $K_d$  secara bertahap hingga nilai 103 yang mana menunjukkan respon yang bagus. Setelah itu ditambahkan nilai  $K_i=0,01$ . Dengan nilai  $K_p = 4$ ,  $K_i = 0,01$ , dan  $K_d = 103$ , papan berhenti bergerak dan seimbang dengan jarak yang terbaca sebesar 28 cm. Respon PID ini memiliki *settling time* sebesar 36,1s dan *steady state error* sebesar 3,7%.

Hasil grafik percobaan ketiga ditunjukkan pada Gambar 11.



Gambar 11. Grafik Percobaan 3 Setpoint 27

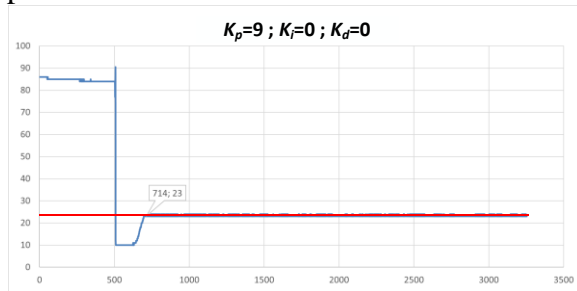
### 3.1.3 Hasil Grafik Setpoint 23

Pada percobaan kali ini, *setpoint* yang ditentukan adalah 33 cm. Pada *setpoint* ini

dilakukan beberapa percobaan nilai  $K_p$ ,  $K_i$ , dan  $K_d$  yang berbeda.

Percobaan 1 : nilai pertama yang diatur adalah nilai  $K_p$  yaitu sebesar 9. Dengan nilai  $K_p=9$ ,  $K_i = 0$ , dan  $K_d = 0$  papan berhenti bergerak dan seimbang dengan jarak yang terbaca sebesar 23 cm dan *settling time* sebesar 7,14s. Respon PID ini memiliki *steady state error* sebesar 0%.

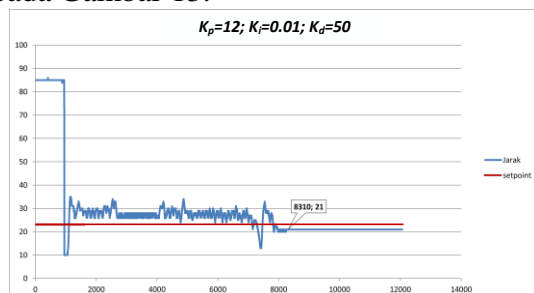
Hasil grafik percobaan pertama ditunjukkan pada Gambar 12.



Gambar 12. Grafik Percobaan 1 Setpoint 23

Percobaan 2 : nilai pertama yang diatur adalah  $K_p = 12$ . Namun, sistem terus berosilasi dan masih jauh dari *setpoint*. Untuk mengurangi *steady state error*, maka ditambahkan nilai  $K_d$  secara bertahap hingga nilai  $K_d=50$ . Setelah itu, ditambahkan nilai  $K_i = 0,01$ . Dengan nilai  $K_p$  12,  $K_i = 0$ , dan  $K_d = 50$ , papan berhenti bergerak dan seimbang dengan jarak yang terbaca sebesar 21cm dan *settling time* sebesar 83s. Namun masih terlihat sedikit osilasi pada grafik setelah papan berhenti. Respon PID ini memiliki *steady state error* sebesar -8,69%.

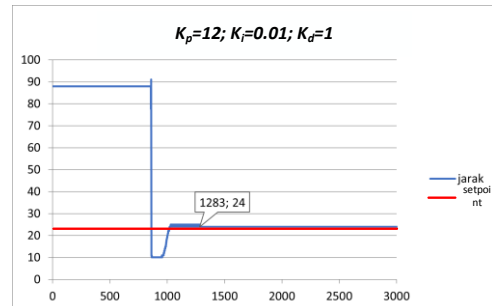
Hasil grafik percobaan kedua ditunjukkan pada Gambar 13.



Gambar 13. Grafik Percobaan 2 Setpoint 23

Percobaan 3 : diberikan nilai awal  $K_p = 12$ . Kemudian, untuk mengurangi nilai *steady state error*, maka ditambahkan nilai  $K_d=1$  dan  $K_i = 0,01$ . Dengan nilai  $K_p = 12$ ,  $K_i = 0,01$ , dan  $K_d = 1$ , papan dalam keadaan seimbang dan bola berhenti bergerak. Respon PID ini memiliki *settling time* sebesar 12,83s dan *steady state error* sebesar 4,3%.

Hasil grafik percobaan ketiga ditunjukkan pada Gambar 14.



Gambar 14. Grafik Percobaan 3 Setpoint 23

### 3.2 Pembahasan

Berikut adalah tabel perbandingan dari hasil percobaan.

Tabel 1. Perbandingan Hasil Percobaan

Hasil Grafik	Nilai Parameter Kendali dan <i>setpoint</i>	Keterangan
Gambar 8	$K_p = 11$ , $K_i = 0,01$ , dan $K_d = 50$ . <i>Setpoint</i> =33.	<i>Setting time</i> = 45 detik. <i>Rata rata Steady state error</i> =3%
Gambar 11	$K_p = 4$ , $K_i = 0,01$ , dan $K_d = 103$ . <i>Setpoint</i> =27.	<i>Setting time</i> = 35 detik. <i>Rata rata Steady state error</i> =4,9%
Gambar 12	$K_p = 9$ , $K_i = 0$ , dan $K_d = 0$ . <i>Setpoint</i> =23.	<i>Setting time</i> = 7,14 detik. <i>Steady state error</i> =4%

#### 3.2.1 *Setpoint* 33 cm

Pada *setpoint* 33 cm, saat percobaan pertama diberikan nilai awal  $K_p=11$ . Namun yang terjadi osilasi yang terlalu lama yang membuat bola terus bergerak. Sehingga nilai  $K_p$  diturunkan menjadi 10. Setelah nilai  $K_p$  diubah maka bola dapat berhenti bergerak. Kemudian dicoba kembali untuk nilai awal  $K_p$  sebesar 10. Didapatkan hasil yang lebih baik dengan *settling time* yang

lebih cepat. Kemudian dicoba kembali dengan  $K_p=11$  namun ditambah dengan  $K_d = 50$ . Dari ketiga hasil tersebut dapat dilihat bahwa untuk *setpoint* 33 cm yang mana posisinya jauh dari titik tengah papan, dibutuhkan kendali  $K_p$  dan atau dengan  $K_d$  untuk membuat bola berhenti dan papan seimbang. Nilai  $K_i$  yang diberikan sangat kecil. Nilai  $K_p$  awal yang diberikan mulai dari 10 dan nilai  $K_d$  yang disesuaikan dengan keadaan respon bola ketika berjalan diatas papan.

### 3.2.2 *Setpoint* 27cm

Pada *setpoint* 27 cm, percobaan pertama dilakukan dengan memberi nilai awal  $K_p=9$ . Namun bola masih bergerak menjauh dari *setpoint*, sehingga nilai  $K_p$  ditambah menjadi 12. Sesaat setelah itu bola dapat berhenti bergerak. Dengan hasil percobaan pertama, maka nilai awal  $K_p$  diturunkan menjadi 9. Pada percobaan kedua terlihat hasil yang lebih baik. Untuk mengurangi *steady state error* maka ditambahkan nilai  $K_d$  saat sistem tersebut masih bekerja dan dengan kendali PD, bola dapat berhenti pada *setpoint*. Pada percobaan ketiga, dicoba kembali dengan nilai awal  $K_d$  yang lebih kecil yaitu 4. Pada nilai  $K_p$  tersebut, sistem sulit mencapai *setpoint* sehingga perlu ditabah nilai  $K_d$ . Sedangkan nilai  $K_i$  bernilai kecil karena dengan kendali PD sistem sudah mampu membuat bola berhenti dan seimbang. Maka, untuk *setpoint* 27 diperlukan kendali PD dengan nilai  $K_p$  awal yan diambil kurang dari atau sama dengan 12.

### 3.2.3 *Setpoint* 23 cm

Pada percobaan pertama untuk *setpoint* 23 cm, nilai awal  $K_p$  sebesar 9 dan hanya dengan kendali P saja bola mampu berhenti dan seimbang. Kemudian dilanjutkan percobaan dengan meningkatkan nilai  $K_p$  yaitu sebesar 12. Dengan nilai  $K_p$  yang lebih besar menghasilkan osilasi yang lebih lama sehingga perlu dikurangi dengan menambahkan nilai  $K_d$ . Dengan kendali PD sistem mampu membuat bola berhenti dan seimbang, namun memiliki *steady state*

*error* yang lebih besar dari percobaan pertama. Kemudian dicoba lagi dengan nilai  $K_p=12$  dan memberikan  $K_d=1$ . Dari nilai tersebut didapatkan hasil *settling time* yang lebih cepat dan *steady state error* yang lebih kecil. Dari hasil tersebut maka didapatkan bahwa untuk *setpoint* 23 cm dibutuhkan nilai  $K_p$  yang kecil yaitu dimulai dari 9 dan nilai  $K_d$  yang kecil pula. Hal ini dikarenakan posisi *setpoint* yang dekat dengan posisi tengah tengah papan. Berdasarkan dari perbandingan beberapa grafik antara tiga *setpoint* yang telah dilakukan, terdapat perbedaan dalam hal nilai nilai parameter yang digunakan. Semakin jauh nilai *setpoint* dari titik tengah papan, maka akan menghasilkan nilai  $K_d$  yang lebih besar. Selain itu, *settling time* untuk *setpoint* yang semakin jauh dari titik tengah papan maka nilai *settling time* akan semakin besar. Artinya, untuk mencapai keadaan *steady state* diperlukan waktu yang lebih lama jika *setpoint* diatur jauh dari titik tengah papan. Rata rata *Steady state error* masih berada dalam batas toleransi yaitu ( $\pm 5\%$ ). Perbedaan hasil pada setiap kendali yang dijalankan dikarenakan gaya gesek papan terhadap bola dan laju pergerakan bola yang berbeda beda pada setiap respon.

## 4. Kesimpulan

Kinerja pengendali PID pada alat peraga keseimbangan bola mampu menyeimbangkan papan dan menghentikan bola pada posisi sesuai *setpoint*. Nilai  $K_p$ ,  $K_i$  dan  $K_d$  yang diberikan sangat memengaruhi respon sistem dalam mencapai *setpoint*. Semakin jauh dari titik tengah papan maka diperlukan nilai  $K_d$  yang lebih besar. Rata rata *error* pada masing masing *setpoint* yaitu pada *setpoint* 33 cm sebesar 3%, pada *setpoint* 27 cm sebesar 4,9% dan pada *setpoint* 23 cm sebesar 4%.

## DAFTAR PUSTAKA

- M. Setiawan. 2012. *Perancangan pengendali pid untuk sistem ball and beam*. Jakarta : Universitas Tri Sakti.



- E. Orlanda Merti Wijaya. 2018. *Perancangan Sistem Keseimbangan Ball and Beam Dengan Menggunakan Pengendali PID Berbasis Arduino Uno*. *J. Tek. Elektro*, vol. 7, no. 01, pp. 53–61.
- H. Muhammad, M. Ramdhani, and A. S. Wibowo. 2017. *Desain dan Implementasi kendali PID pada Beam and Ball System*. *e-Proceeding Eng.*, vol. 4, no. 3, pp. 3286–3293.
- M. Saad and M. Khalfallah. 2017. *Design and Implementation of an Embedded Ball-beam Controller Using PID Algorithm*. *Univers. J. Control Autom.*, vol. 5, no. 4, pp. 63–70.
- B. Supriyo, S. B. Kuntardjo, and Sihono. 2017. *Alat Peraga Kendali Pemanas Udara Berbasis Arduino Uno Sebagai Penunjang Praktikum Laboratorium Kendali*. *J. Sains dan Teknol.*, vol. 1, no. 1, pp. 1–14.