

BAB V

PENGUJIAN DAN ANALISIS SISTEM

Tujuan pengujian sistem ini adalah untuk menentukan apakah alat yang telah dibuat berfungsi dengan baik dan sesuai dengan perancangan. Pengujian pada sistem ini meliputi pengujian setiap blok maupun pengujian secara keseluruhan. Pengujian setiap blok ini dilakukan untuk menemukan letak kesalahan dan mempermudah analisis pada sistem apabila alat tidak bekerja sesuai dengan perancangan. Dan, pengujian yang dilakukan adalah sebagai berikut:

1. Pengujian *Rotary Encoder*
2. Pengujian Motor DC
3. Pengujian *Magnetic Encoder*
4. Pengujian *Driver Motor*
5. Pengujian secara keseluruhan

5.1. Pengujian *Rotary Encoder*

a. Tujuan

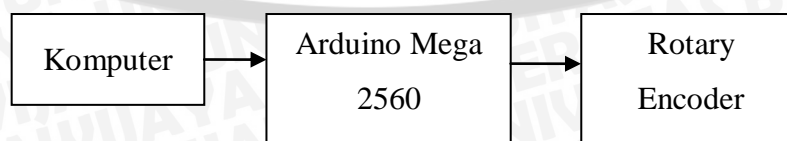
Mengetahui perubahan sudut dengan *counter* pada *rod* dari *Rotary Inverted Pendulum*.

b. Peralatan yang digunakan:

- Sistem mekanik *Rotary Inverted Pendulum* dengan *rotary encoder* terhubung dengan *rod* dari *Rotary Inverted Pendulum*.
- Komputer
- Arduino Mega 2560
- Program Arduino

c. Langkah Pengujian:

1. Menyusun alat seperti dalam Gambar 5.1.



Gambar 5.1 Diagram Blok Pengujian *Rotary Encoder*

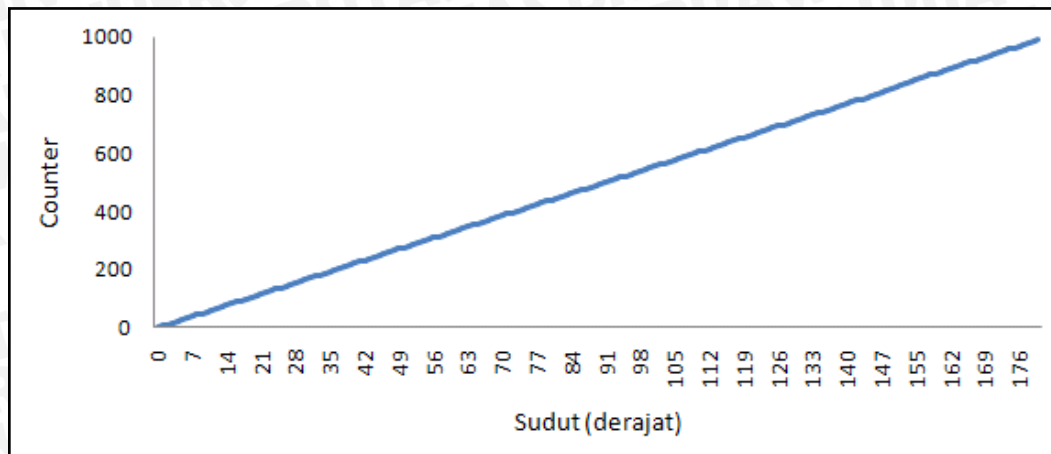
2. Mengunggah program pembacaan sensor *rotary encoder* ke Arduino Mega 2560.
3. Mengubah kemiringan *rod* dari *Rotary Inverted Pendulum* yang telah terhubung dengan *rotary encoder*, mulai dari sudut 0° hingga 180° .
4. Menggambar data perbandingan *counter* dengan sudut keluaran *rotary encoder* pada grafik untuk melihat kelinieran dari *rotary encoder*.

Dan data hasil pengujian *rotary encoder* pada *rod* ditunjukkan dalam Tabel 5.1, dan grafik hubungan antara sudut dan *counter rotary encoder* dapat dilihat pada Gambar 5.2.

Tabel 5.1 Hasil Pengujian *Rotary Encoder* pada *Rod Pendulum*

Counter	Derajat	Counter	Derajat	Counter	Derajat	Counter	Derajat
0	0	110	20	220	40	330	60
5	1	115	21	226	41	336	61
10	2	120	22	231	42	341	62
16	3	126	23	236	43	346	63
22	4	132	24	242	44	352	64
27	5	137	25	248	45	358	65
32	6	142	26	253	46	363	66
38	7	148	27	258	47	368	67
44	8	154	28	264	48	374	68
49	9	159	29	270	49	380	69
54	10	164	30	275	50	385	70
60	11	170	31	280	51	390	71
66	12	176	32	286	52	396	72
71	13	181	33	292	53	402	73
76	14	186	34	297	54	407	74
82	15	192	35	302	55	412	75
88	16	198	36	308	56	418	76
93	17	203	37	314	57	424	77
98	18	208	38	319	58	429	78
104	19	214	39	324	59	434	79

Counter	Derajat	Counter	Derajat	Counter	Derajat	Counter	Derajat
440	80	578	105	716	130	858	156
446	81	583	106	721	131	864	157
451	82	588	107	726	132	870	158
456	83	594	108	732	133	875	159
462	84	600	109	738	134	880	160
468	85	605	110	743	135	886	161
473	86	610	111	748	136	892	162
478	87	616	112	754	137	897	163
484	88	622	113	760	138	902	164
490	89	627	114	765	139	908	165
495	90	632	115	770	140	914	166
500	91	638	116	776	141	919	167
506	92	644	117	782	142	924	168
512	93	649	118	787	143	930	169
517	94	654	119	792	144	936	170
522	95	660	120	798	145	941	171
528	96	666	121	804	146	946	172
534	97	671	122	809	147	952	173
539	98	676	123	814	148	958	174
544	99	682	124	820	149	963	175
550	100	688	125	831	151	968	176
556	101	693	126	836	152	974	177
561	102	698	127	842	153	980	178
566	103	704	128	848	154	985	179
572	104	710	129	853	155	990	180



Gambar 5.2 Grafik Hubungan antara Sudut dan *Counter Rotary Encoder* pada *Rod Pendulum*

Dari hasil pengujian yang dilakukan pada *rotary encoder* dapat dilihat bahwa *rotary encoder* yang digunakan mempunyai kelinieran yang baik, sehingga ideal untuk digunakan sebagai pendeteksi kemiringan dari *rod* pada *Rotary Inverted Pendulum*.

5.2. Pengujian Motor DC

a. Tujuan

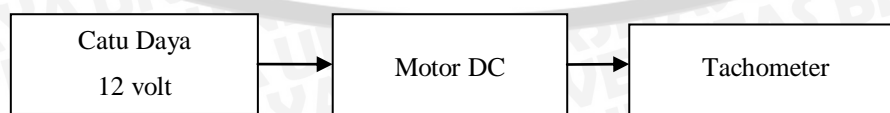
Mengetahui karakteristik motor DC yang digunakan pada sistem *Rotary Inverted Pendulum* terhadap tegangan.

b. Peralatan yang digunakan:

- Motor DC
- Catu daya 12 volt
- *Rotary Encoder*
- Tachometer
- Multimeter

c. Langkah Pengujian:

1. Menyusun alat seperti dalam Gambar 5.3.



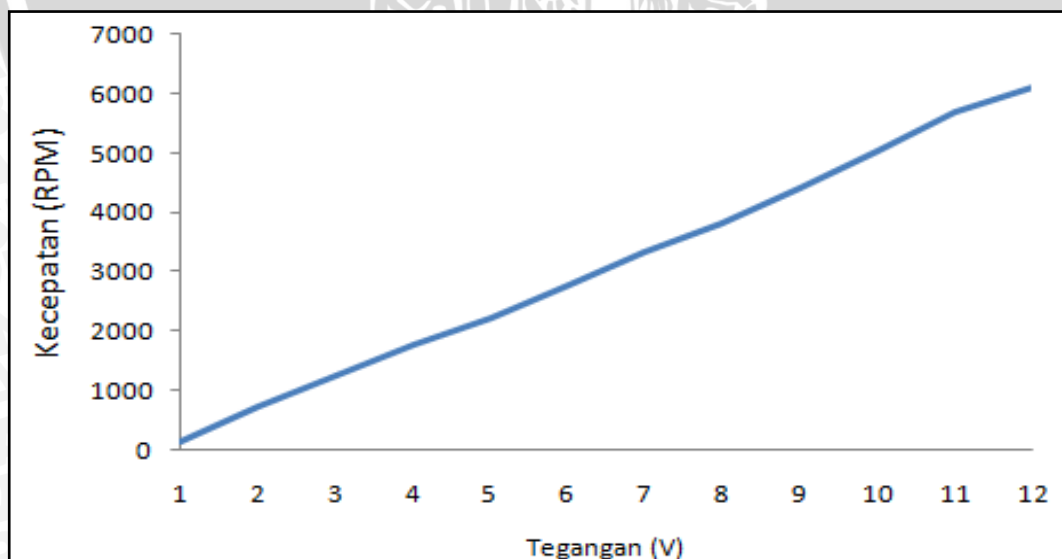
Gambar 5.3 Diagram Blok Pengujian Motor DC

2. Mengatur tegangan *input* motor mulai dari 0 volt hingga 12 volt dengan *interval* 1 volt.
3. Mengukur kecepatan motor dengan tachometer.
4. Menggambar data kecepatan motor untuk mendapatkan grafik karakteristik kecepatan motor terhadap tegangan.

Tabel 5.2 menunjukkan data dari pengujian motor, dan grafik karakteristik motor DC ditunjukkan dalam Gambar 5.4.

Tabel 5.2 Hasil Pengujian Motor DC

Tegangan (V)	Kecepatan (RPM)
1	147
2	730
3	1259
4	1768
5	2206
6	2783
7	3329
8	3806
9	4387
10	5042
11	5676
12	6098



Gambar 5.4 Gambar Hubungan antara Tegangan Motor terhadap Kecepatan Motor

Dari hasil pengujian dapat dilihat bahwa motor berputar dengan tegangan minimal sebesar 1 volt. Selain itu, dapat dilihat bahwa hubungan antara tegangan dan kecepatan motor relatif linier. Sehingga dapat disimpulkan bahwa motor DC ini cukup baik untuk diterapkan pada sistem *Rotary Inverted Pendulum*.

5.3. Pengujian *Magnetic Encoder*

a. Tujuan

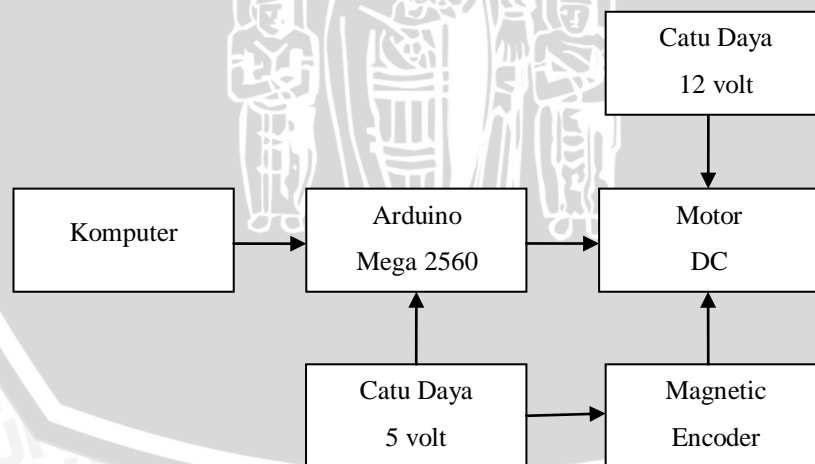
Mengetahui kecepatan motor DC yang terbaca pada sensor *magnetic encoder* telah sesuai dengan yang diinginkan.

b. Peralatan yang digunakan:

- Motor DC
- Catu daya 12 volt
- *Magnetic Encoder*
- Komputer
- Arduino Mega 2560
- Program Arduino

c. Langkah Pengujian:

1. Menyusun alat seperti dalam Gambar 5.5.



Gambar 5.5 Diagram Blok Pengujian *Magnetic Encoder*

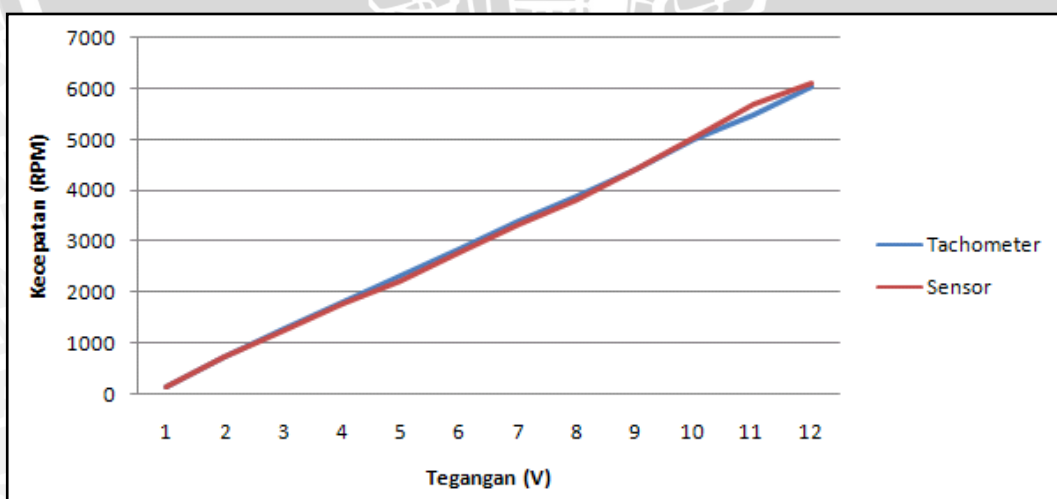
2. Mengunduh Arduino Mega 2560 dengan program pembacaan sensor *magnetic encoder*.

3. Mengatur tegangan *input* motor mulai dari 0 volt hingga 12 volt dengan *interval* 1 volt.
4. Mencatat data kecepatan motor yang diperoleh dari pembacaan sensor *magnetic encoder*.

Tabel 5.3 menunjukkan data dari pengujian sensor *magnetic encoder*, dan Gambar 5.6 merupakan grafik perbandingan pembacaan kecepatan motor DC antara sensor *magnetic encoder* dengan tachometer.

Tabel 5.3 Hasil Pengujian Sensor *Magnetic Encoder* dalam Membaca Kecepatan Motor DC

Tegangan (V)	Kecepatan (RPM)	
	<i>Magnetic Encoder</i>	Tachometer
1	150.3	147
2	742.8	730
3	1311.3	1259
4	1826	1768
5	2331.2	2206
6	2866.2	2783
7	3413.2	3329
8	3900.5	3806
9	4415.2	4387
10	5002.8	5042
11	5509.2	5676
12	6051.4	6098



Gambar 5.6 Grafik Perbandingan Pembacaan Kecepatan Motor DC antara Sensor dengan Tachometer

Dari hasil pengujian yang dilakukan dapat disimpulkan bahwa hasil pengujian sensor *magnetic encoder* sesuai dengan data pengukuran secara praktek menggunakan tachometer dan sensor mampu berjalan dengan baik.

5.4. Pengujian *Driver Motor*

a. Tujuan

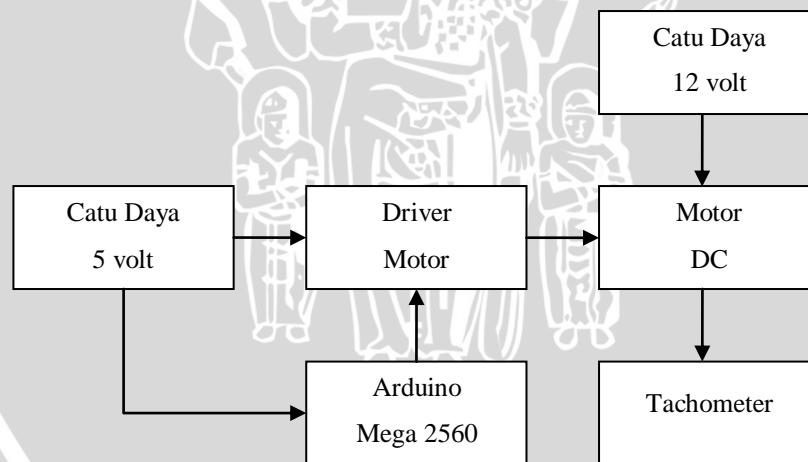
Mengetahui *output* dari *driver* motor apabila diberi *input* yang berbeda-beda.

b. Peralatan yang digunakan:

- *Driver* motor EMS 5A *H-Bridge*
- Arduino Mega 2560
- Kabel serial
- Motor DC
- Catu daya 5V dan 12V

c. Langkah Pengujian:

1. Menyusun alat seperti dalam Gambar 5.7.



Gambar 5.7 Diagram Blok Pengujian *Driver Motor*

2. Membuat program untuk menentukan arah dan kecepatan motor pada *software* arduino.
3. Mengukur kecepatan putar motor dengan tachometer.
4. Menggambar data yang didapatkan pada grafik.

Tabel 5.4 menunjukkan data pengujian *driver* motor untuk arah putaran motor.

Tabel 5.4 Hasil Pengujian Arah Putaran Motor

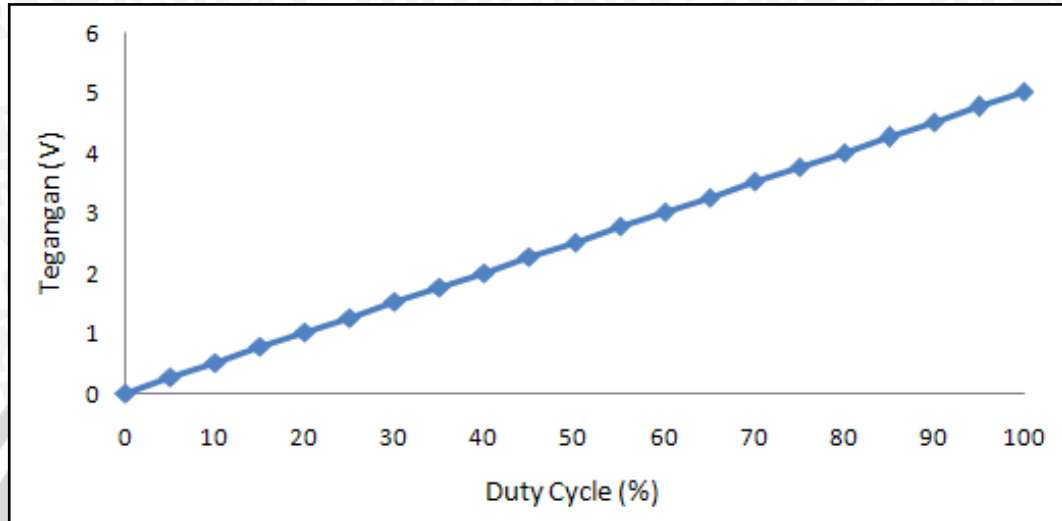
Tegangan pada Pin (V)						Arah Putar Motor
MEN	MEN	MEN	MEN	MEN	MEN	
5	5	5	0	12	0	Putar Kanan
0	5	0	0	0	0	Berhenti
5	5	0	5	0	12	Putar Kiri
5	0	0	0	0	0	Berhenti

Seperti yang telah dijelaskan pada perancangan perangkat keras, ada empat buah *pin input* pada *driver* motor. Empat buah *pin input* tersebut adalah *pin* MIN1 dan MIN2 dari *driver* motor dan berfungsi sebagai penentu arah berputarnya motor, *pin* MEN dari *driver* motor yang berfungsi sebagai penentu kecepatan motor, serta *pin* MSLP untuk mengatur kerja *driver* motor. Hasil pengujian didapatkan seperti dalam Tabel 5.5.

Tabel 5.5 Hasil Pengujian Motor dengan PWM

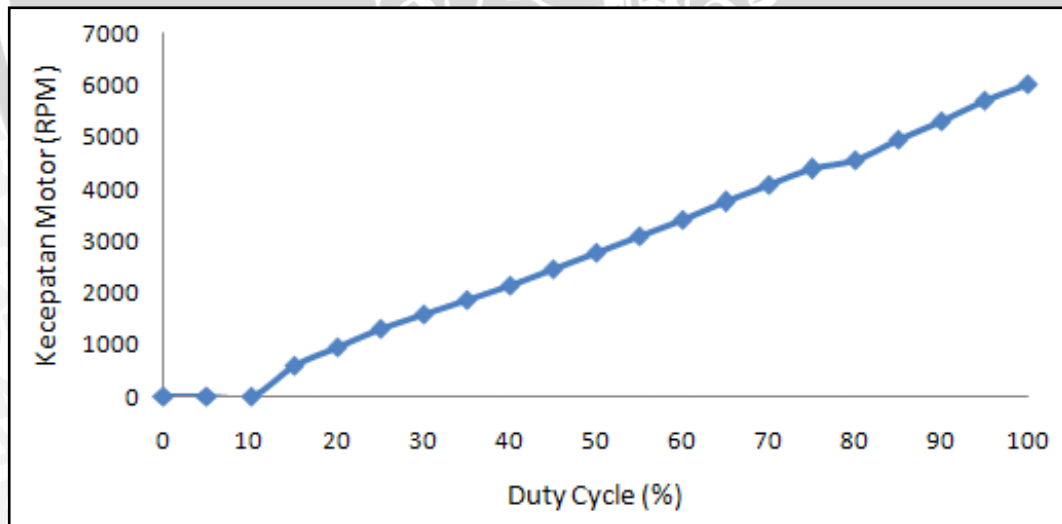
MIN 1 (%)	MIN 2 (%)	Tegangan pada Pin (V)			Kecepatan Motor (RPM)
		MEN	MOUT 1	MOUT2	
0	100	0	0	0	0
5	100	0,282	4,3	0	0
10	100	0,518	4,69	0	0
15	100	0,774	5,12	0	607
20	100	1,029	5,54	0	959
25	100	1,265	5,93	0	1300
30	100	1,52	6,35	0	1576
35	100	1,776	6,78	0	1853
40	100	2,011	7,17	0	2149
45	100	2,267	7,59	0	2457
50	100	2,523	8,01	0	2775
55	100	2,779	8,4	0	3090
60	100	3,015	8,79	0	3404
65	100	3,27	9,2	0	3743
70	100	3,525	9,62	0	4081
75	100	3,781	10,004	0	4383
80	100	4,016	10,23	0	4561
85	100	4,27	10,84	0	4962
90	100	4,53	11,26	0	5323
95	100	4,78	11,67	0	5711
100	100	5,04	12,03	0	6035

Pada Gambar 5.8 dapat dilihat grafik hubungan antara tegangan *enable pin* MEN *driver* motor dengan *duty cycle* PWM (%) pada Arduino. Dari grafik dapat dilihat bahwa hubungan antara kedua nilai tersebut linier.

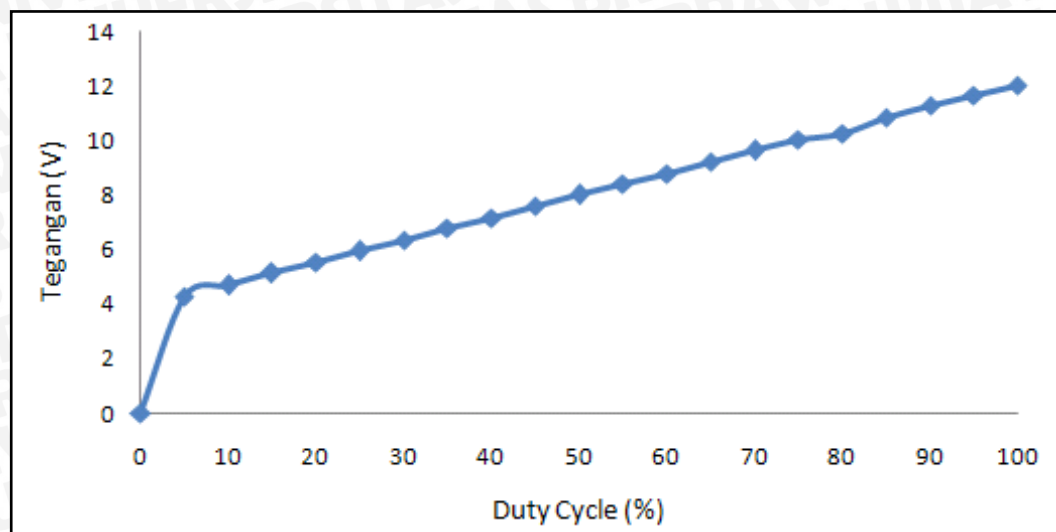


Gambar 5.8 Hubungan antara *Duty Cycle* PWM dengan Tegangan *Enable Driver* Motor

Hubungan antara *duty cycle* PWM (%) terhadap kecepatan motor dapat dilihat pada Gambar 5.9, dan hubungan antara *duty cycle* PWM (%) terhadap tegangan dapat dilihat pada Gambar 5.10.



Gambar 5.9 Hubungan antara *Duty Cycle* PWM dengan Kecepatan Motor



Gambar 5.10 Hubungan antara *Duty Cycle* dengan Tegangan Motor

Berdasarkan hasil pengujian *driver* motor dapat dilihat bahwa motor akan mulai bergerak ketika *duty cycle* bernilai 15%. Tegangan *enable*, tegangan motor, dan *duty cycle* memiliki hubungan yang linier. Berdasarkan data-data tersebut *driver* motor EMS 5A *H-Bridge* memiliki kemampuan yang baik untuk men-*drive* motor DC.

5.5. Pengujian Keseluruhan

a. Tujuan

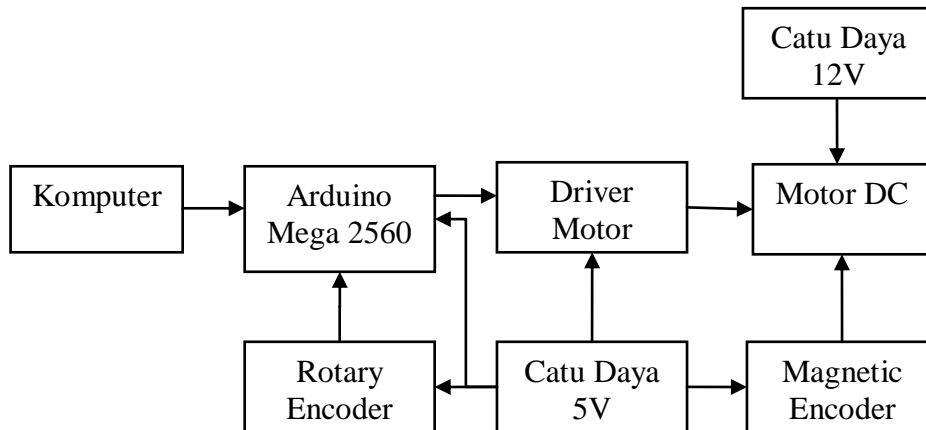
Mengetahui kerja dari perangkat keras dan perangkat lunak setelah diintegrasikan dalam sebuah sistem terpadu.

b. Peralatan yang digunakan:

- *Plant Rotary Inverted Pendulum* lengkap dengan *rotary encoder*, *magnetic encoder*, dan motor DC.
- *Driver* motor EMS 5A *H-Bridge*
- Arduino Mega 2560
- Kabel serial

c. Langkah Pengujian:

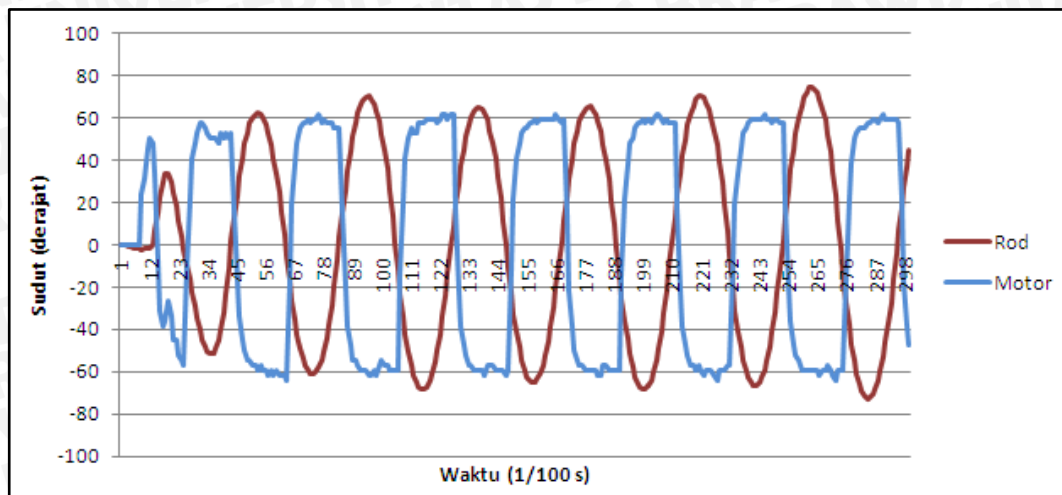
1. Menyusun alat seperti pada Gambar 5.11.



Gambar 5.11 Diagram Blok Pengujian Keseluruhan

2. Mengaktifkan semua catu daya.
3. Membuat program untuk pengendalian *Rotary Inverted Pendulum* dengan kontroler PID pada *software* Arduino.
4. Mengamati dan menganalisa hasil dari kinerja *Rotary Inverted Pendulum*.

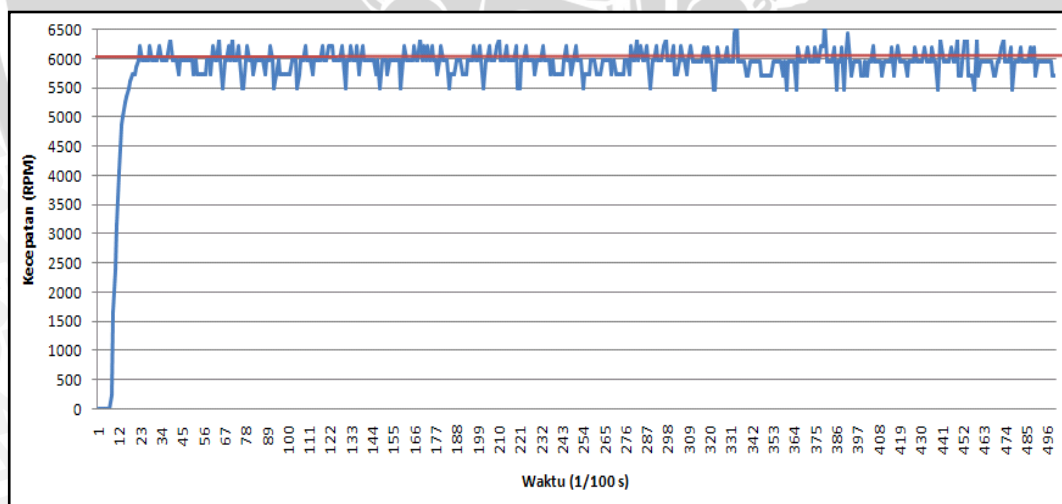
Untuk melakukan pengujian ini, mula-mula *rod* pendulum berada pada posisi bawah (*downright position*), kemudian *rod* pendulum diberikan gangguan agar *plant* dapat merespon untuk melakukan gerakan *Swing-Up*. Motor DC akan mengayunkan *rod* pendulum ke kiri dan ke kanan hingga *rod* pendulum mencapai pada posisi tegak lurus (*upright position*). Gambar 5.12 menunjukkan grafik respon dari *Rotary Inverted Pendulum* untuk gerakan *Swing-Up*.



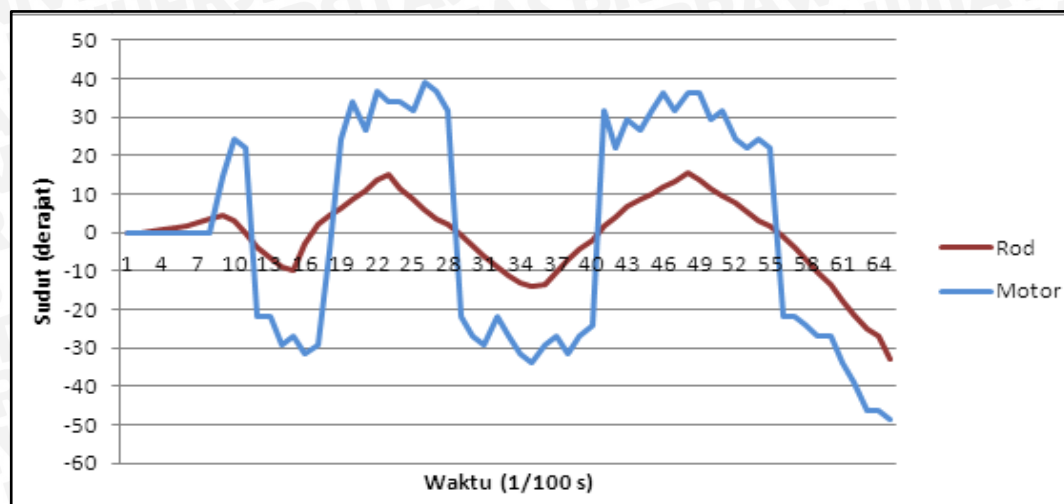
Gambar 5.12. Grafik Respon *Rotary Inverted Pendulum* untuk Gerakan *Swing-Up*

Pada proses *Swing-Up*, *rod* pendulum tidak bisa mencapai posisi tegak lurus (*upright position*) dikarenakan adanya kendala mekanis pada alat.

Selanjutnya, dilakukan pengujian untuk proses penstabilan lengan pendulum dengan nilai parameter PID yang telah diperoleh sebelumnya, yaitu $K_p=0.73$, $K_i=0.041$, dan $K_d=3.285$. Gambar 5.13 menunjukkan grafik respon dari motor DC dan Gambar 5.14 menunjukkan grafik respon dari *Rotary Inverted Pendulum* dengan nilai PID tersebut.



Gambar 5.13 Grafik Respon Motor DC dengan nilai $K_p=0.73$, $K_i=0.041$, dan $K_d=3.285$



Gambar 5.14 Grafik Respon *Rotary Inverted Pendulum* dengan nilai $K_p=0.73$, $K_i=0.041$, dan $K_d=3.285$

Dari Gambar 5.13 dan Gambar 5.14 terlihat bahwa respon motor DC dan *Rotary Inverted Pendulum* dengan nilai PID $K_p=0.73$, $K_i=0.041$, dan $K_d=3.285$ masih kurang baik, sehingga perlu dilakukan *tuning* kembali menggunakan *hand tuning* untuk memperoleh hasil respon yang lebih baik.

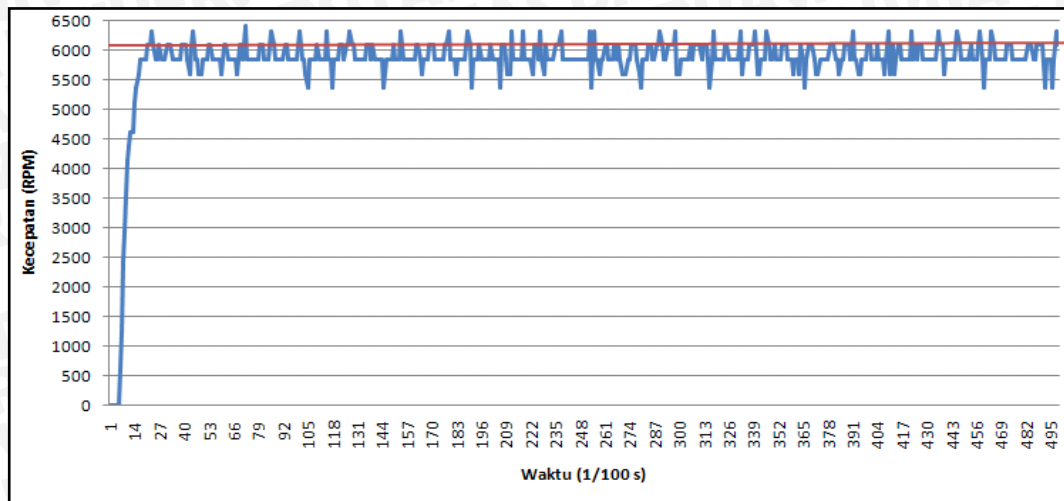
Menurut Smith (1979), untuk melakukan *hand tuning* prosedur yang dilakukan adalah sebagai berikut :

- 1) Melepaskan kontroler integral dan diferensial dengan memberikan nilai $K_i=0$ dan $K_d=0$.
- 2) Mengatur nilai K_p hingga didapatkan respon yang diinginkan, dengan mengabaikan *offset* dari *setpoint*.
- 3) Dengan terus menaikkan nilai K_p , nilai dari K_d dinaikkan untuk mengurangi *overshoot* yang terjadi.
- 4) Naikkan nilai K_i untuk mengurangi *offset*.

Dengan menggunakan metode *hand tuning*, nilai parameter PID perlu diubah-ubah secara *trial* dan *error* agar respon yang diperoleh sesuai dengan harapan.

Proses *tuning* parameter PID ini dilakukan dengan cara mengatur nilai K_p hingga didapatkan respon motor DC yang mendekati *setpoint* 6098 rpm.

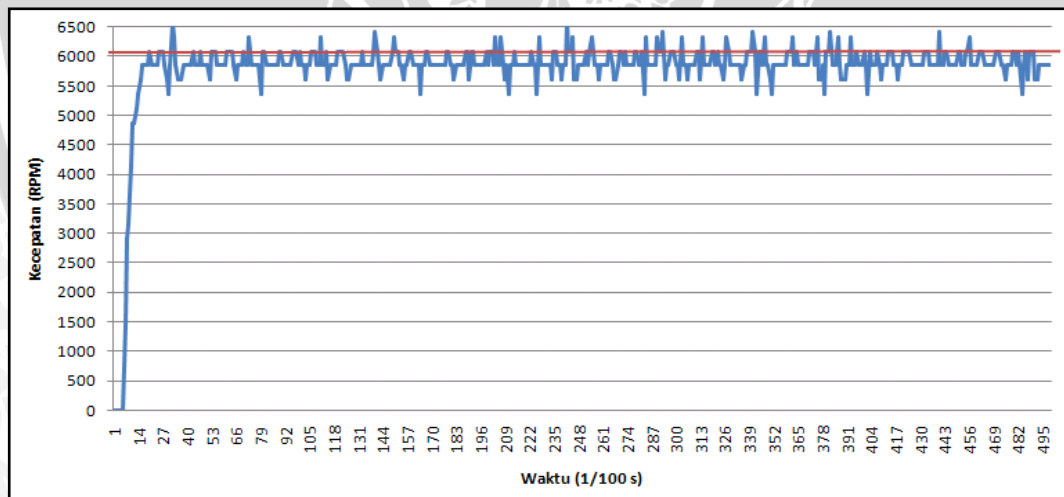
Untuk nilai $K_p=0.95$, $K_i=0$, dan $K_d=0$ diperoleh grafik respon motor DC seperti pada Gambar 5.15.



Gambar 5.15 Grafik Respon Motor DC dengan nilai $K_p=0.95$, $K_i=0$, dan $K_d=0$

$$\begin{aligned} \%Ess &= \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100\% \\ &= \frac{|5923.492 - 6098|}{6098} \times 100\% \\ &= 2.86\% \text{ dengan waktu } steady \text{ 0.25 detik.} \end{aligned}$$

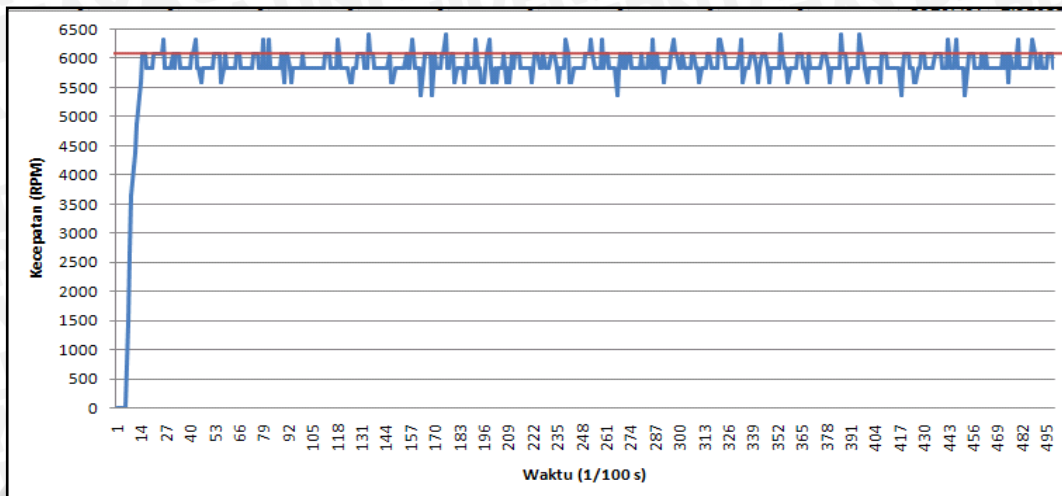
Untuk nilai $K_p=1$, $K_i=0$, dan $K_d=0$ diperoleh grafik respon motor DC seperti pada Gambar 5.16.



Gambar 5.16 Grafik Respon Motor DC dengan nilai $K_p=1$, $K_i=0$, dan $K_d=0$

$$\begin{aligned} \%Ess &= \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100\% \\ &= \frac{|5924.389 - 6098|}{6098} \times 100\% \\ &= 2.84\% \text{ dengan waktu } steady \text{ 0.24 detik.} \end{aligned}$$

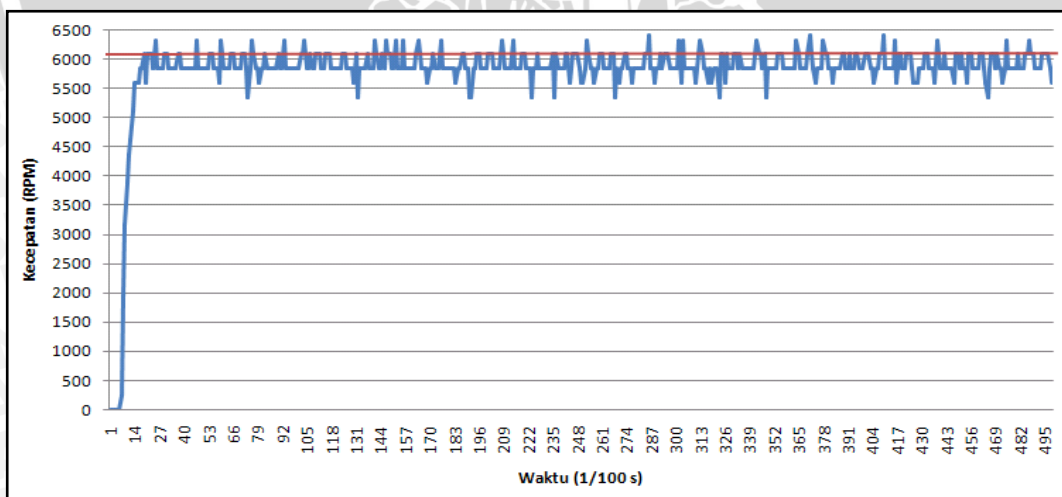
Untuk nilai $K_p=1.05$, $K_i=0$, dan $K_d=0$ diperoleh grafik respon motor DC seperti pada Gambar 5.17.



Gambar 5.17 Grafik Respon Motor DC dengan nilai $K_p=1.05$, $K_i=0$, dan $K_d=0$

$$\begin{aligned} \%Ess &= \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100\% \\ &= \frac{|5927.424 - 6098|}{6098} \times 100\% \\ &= 2.80\% \text{ dengan waktu } steady\ 0.23 \text{ detik.} \end{aligned}$$

Untuk nilai $K_p=1.1$, $K_i=0$, dan $K_d=0$ diperoleh grafik respon motor DC seperti pada Gambar 5.18.



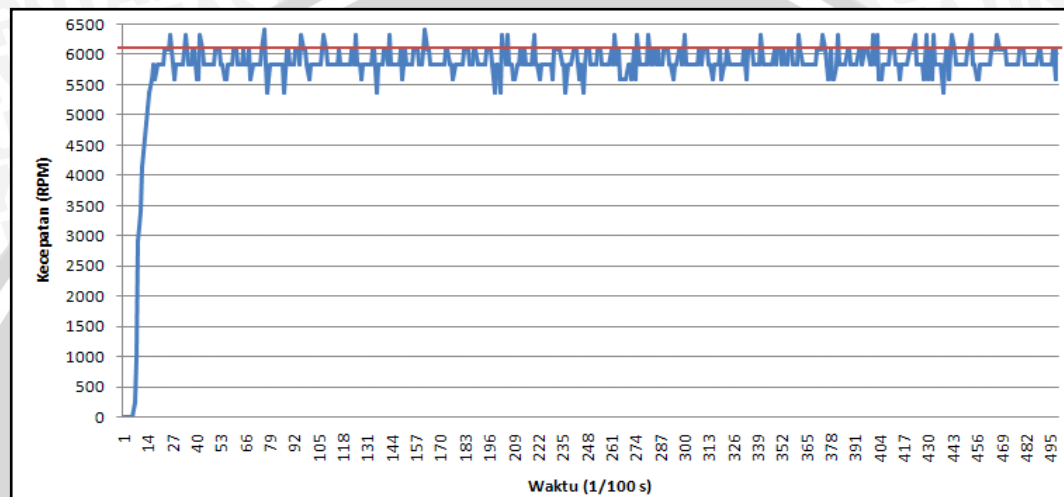
Gambar 5.18 Grafik Respon Motor DC dengan nilai $K_p=1.1$, $K_i=0$, dan $K_d=0$

$$\%Ess = \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100\%$$

$$= \frac{|5926.369 - 6098|}{6098} \times 100\%$$

= 2.81% dengan waktu *steady* 0.23 detik.

Untuk nilai $K_p=1.15$, $K_i=0$, dan $K_d=0$ diperoleh grafik respon motor DC seperti pada Gambar 5.19.



Gambar 5.19 Grafik Respon Motor DC dengan nilai $K_p=1.15$, $K_i=0$, dan $K_d=0$

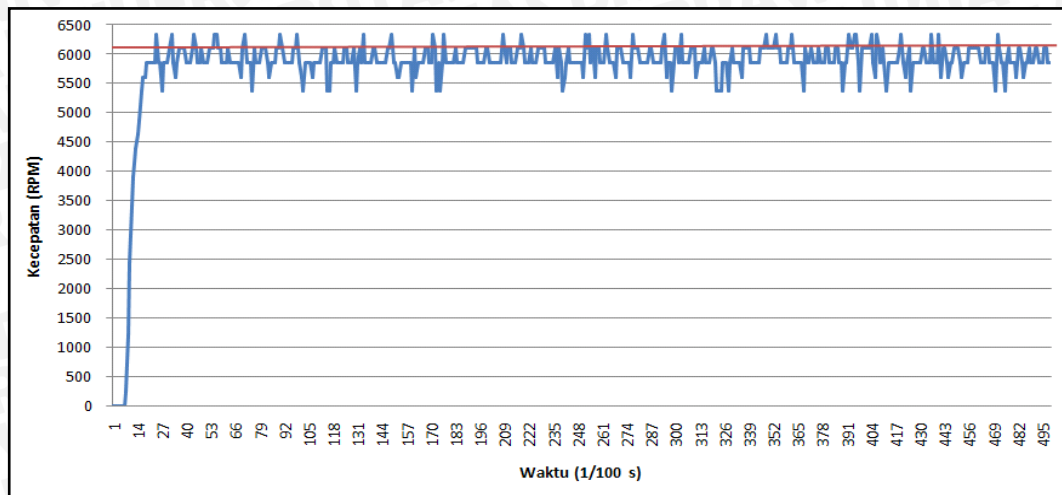
$$\%Ess = \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100\%$$

$$= \frac{|5923.719 - 6098|}{6098} \times 100\%$$

= 2.86% dengan waktu *steady* 0.24 detik.

Dari hasil perancangan tersebut, nilai $K_p=1.05$ dapat mencapai *steady* dan memiliki *error steady state* paling kecil walaupun masih terjadi *overshoot* dari titik *steady* sekarang. Setelah diperoleh nilai K_p yang cukup baik, untuk mengurangi *overshoot* maka digunakanlah kontroler *derivative*.

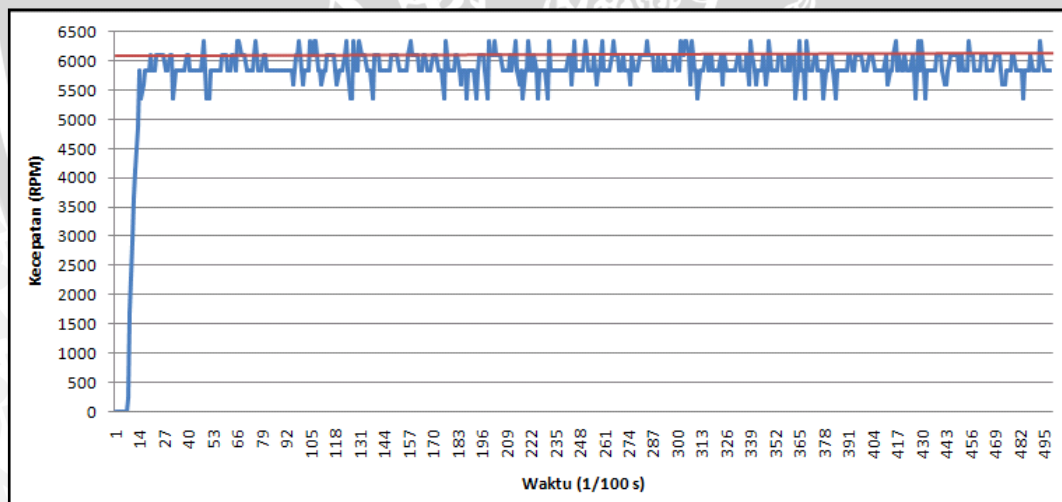
Untuk nilai $K_p=1.05$, $K_i=0$, dan $K_d=5.5$ diperoleh grafik respon motor DC seperti pada Gambar 5.20.



Gambar 5.20 Grafik Respon Motor DC dengan nilai $K_p=1.05$, $K_i=0$, dan $K_d=5.5$

$$\begin{aligned} \%Ess &= \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100\% \\ &= \frac{|5932.412 - 6098|}{6098} \times 100\% \\ &= 2.71\% \text{ dengan waktu } steady \text{ 0.23 detik.} \end{aligned}$$

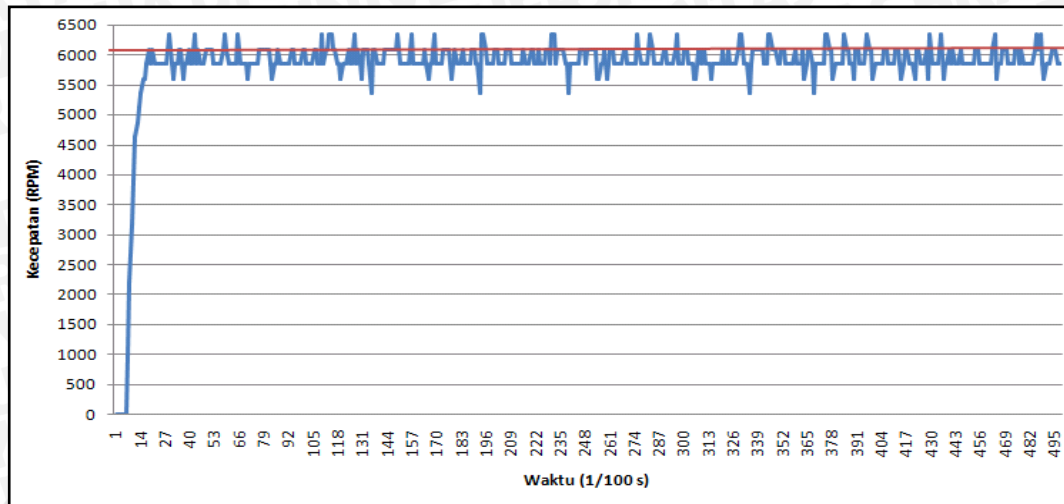
Untuk nilai $K_p=1.05$, $K_i=0$, dan $K_d=6$ diperoleh grafik respon motor DC seperti pada Gambar 5.21.



Gambar 5.21 Grafik Respon Motor DC dengan nilai $K_p=1.05$, $K_i=0$, dan $K_d=6$

$$\begin{aligned} \%Ess &= \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100\% \\ &= \frac{|5938.505 - 6098|}{6098} \times 100\% \\ &= 2.62\% \text{ dengan waktu } steady \text{ 0.22 detik.} \end{aligned}$$

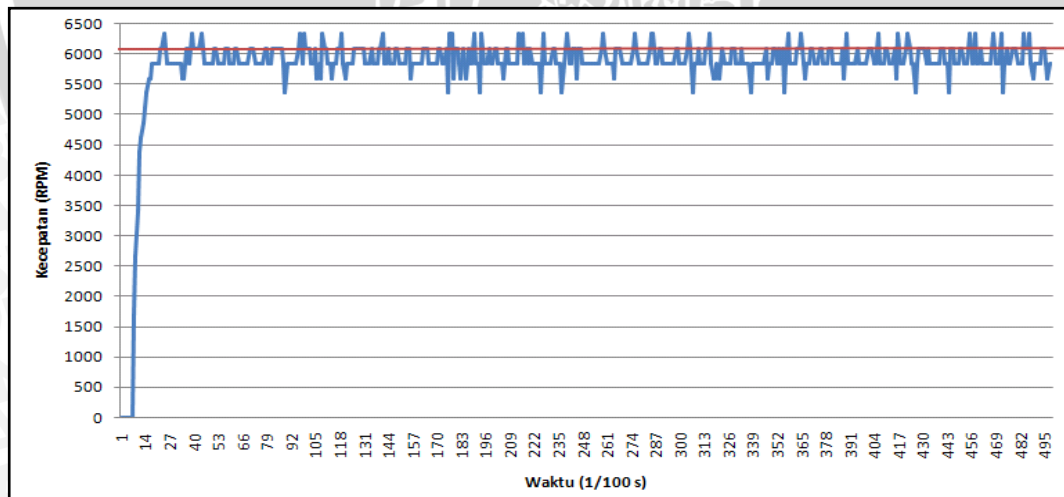
Untuk nilai $K_p=1.05$, $K_i=0$, dan $K_d=6.5$ diperoleh grafik respon motor DC seperti pada Gambar 5.22.



Gambar 5.22 Grafik Respon Motor DC dengan nilai $K_p=1.05$, $K_i=0$, dan $K_d=6.5$

$$\begin{aligned} \%ESS &= \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100\% \\ &= \frac{|5951.885 - 6098|}{6098} \times 100\% \\ &= 2.40\% \text{ dengan waktu } steady\ 0.20 \text{ detik.} \end{aligned}$$

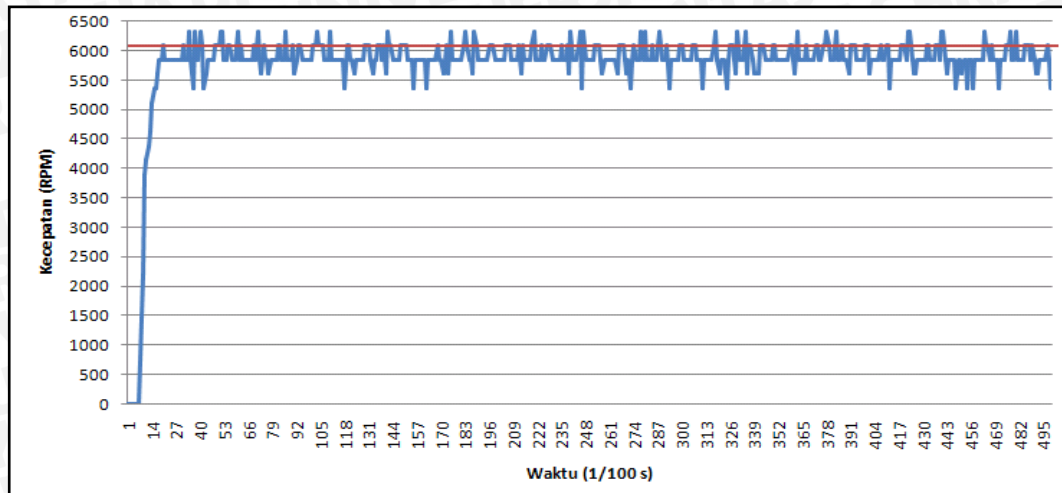
Untuk nilai $K_p = 1.05$, $K_i = 0$, dan $K_d = 7$ diperoleh grafik respon motor DC seperti pada Gambar 5.23.



Gambar 5.23 Grafik Respon Motor DC dengan nilai $K_p=1.05$, $K_i=0$, dan $K_d=7$

$$\begin{aligned} \%ESS &= \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100\% \\ &= \frac{|5940.929 - 6098|}{6098} \times 100\% \\ &= 2.56\% \text{ dengan waktu } steady\ 0.21 \text{ detik.} \end{aligned}$$

Untuk nilai $K_p=1.05$, $K_i=0$, dan $K_d=7.5$ diperoleh grafik respon motor DC seperti pada Gambar 5.24.

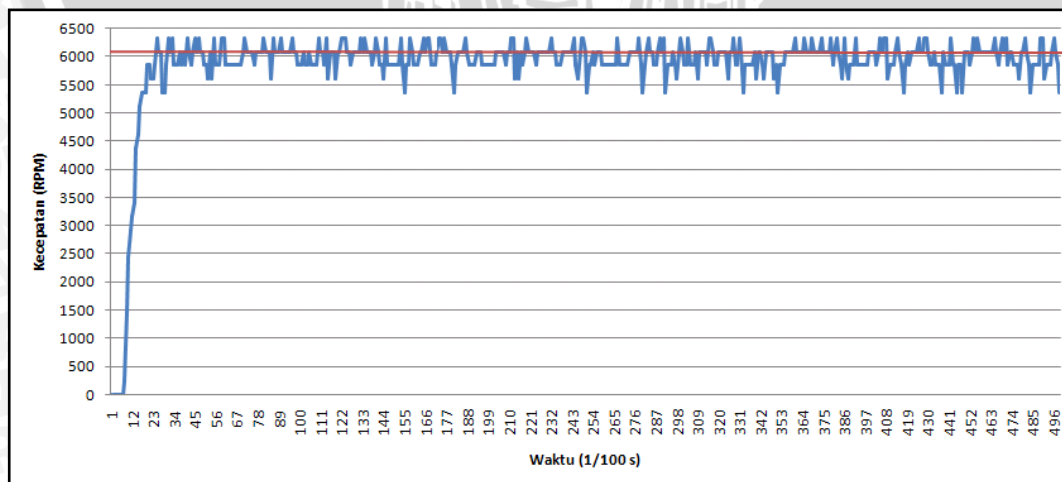


Gambar 5.24 Grafik Respon Motor DC dengan nilai $K_p=1.05$, $K_i=0$, dan $K_d=7.5$

$$\begin{aligned} \%Ess &= \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100\% \\ &= \frac{|5928.751 - 6098|}{6098} \times 100\% \\ &= 2.77\% \text{ dengan waktu } steady \text{ 0.23 detik.} \end{aligned}$$

Dari hasil perancangan tersebut, menunjukkan nilai $K_d=6.5$ mengurangi *overshoot* tetapi masih terdapat *offset*. Untuk menghilangkan *offset* yang terjadi maka digunakanlah kontroler integral.

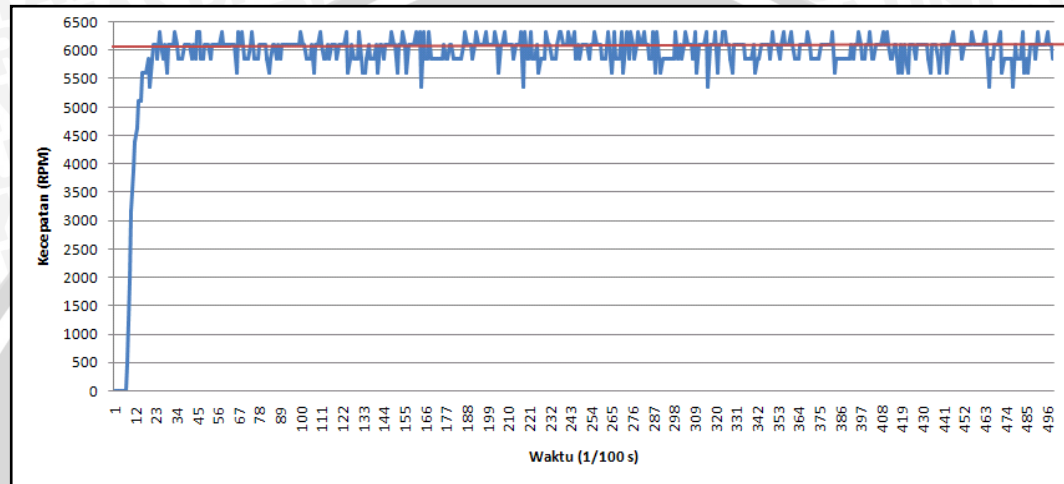
Untuk nilai $K_p = 1.05$, $K_i = 0.02$, dan $K_d = 6.5$ diperoleh grafik respon seperti pada Gambar 5.25.



Gambar 5.25 Grafik Respon Motor DC dengan nilai $K_p=1.05$, $K_i=0.02$, dan $K_d=6.5$

$$\begin{aligned} \%ESS &= \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100\% \\ &= \frac{|6014.634 - 6098|}{6098} \times 100\% \\ &= 1.37\% \text{ dengan waktu } steady \text{ 0.24 detik.} \end{aligned}$$

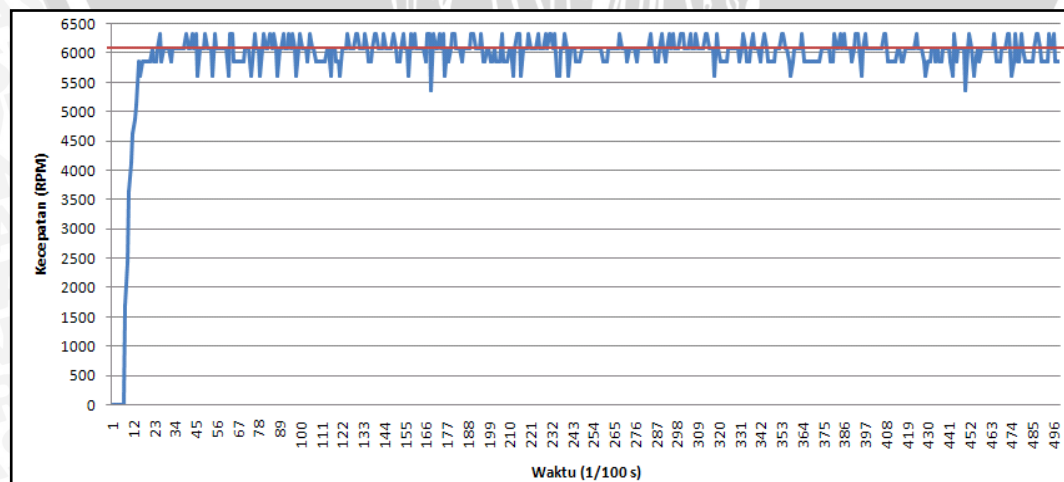
Untuk nilai $K_p = 1.05$, $K_i = 0.025$, dan $K_d = 6.5$ diperoleh grafik respon seperti pada Gambar 5.26.



Gambar 5.26 Grafik Respon Motor DC dengan nilai $K_p=1.05$, $K_i=0.025$, dan $K_d=6.5$

$$\begin{aligned} \%ESS &= \frac{|Averagespeedsteady - setpoint|}{setpoint} \times 100\% \\ &= \frac{|6030.465 - 6098|}{6098} \times 100\% \\ &= 1.11\% \text{ dengan waktu } steady \text{ 0.23 detik.} \end{aligned}$$

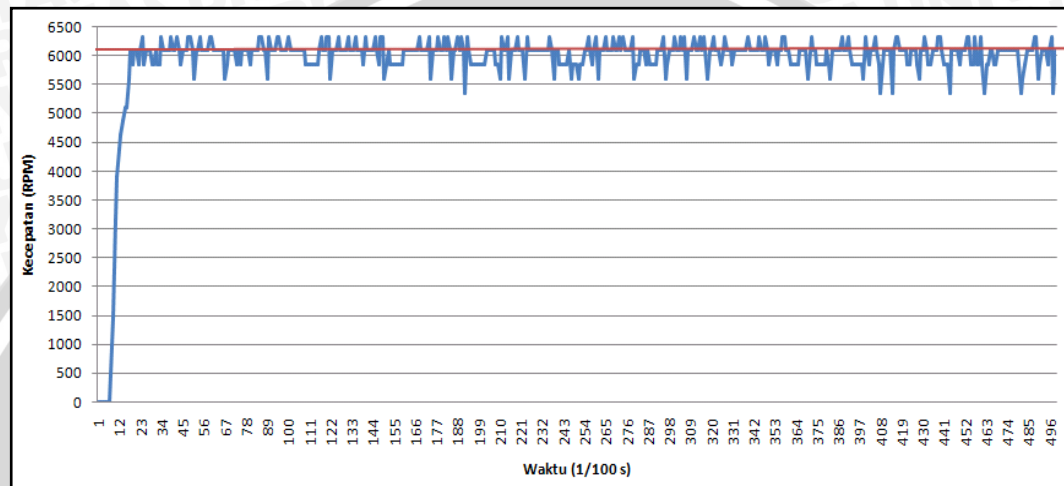
Untuk nilai $K_p = 1.05$, $K_i = 0.03$, dan $K_d = 6.5$ diperoleh grafik respon seperti pada Gambar 5.27.



Gambar 5.27 Grafik Respon Motor DC dengan nilai $K_p=1.05$, $K_i=0.03$, dan $K_d=6.5$

$$\begin{aligned} \%Ess &= \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100\% \\ &= \frac{|6071.877 - 6098|}{6098} \times 100\% \\ &= 0.43\% \text{ dengan waktu } steady \text{ 0.20 detik.} \end{aligned}$$

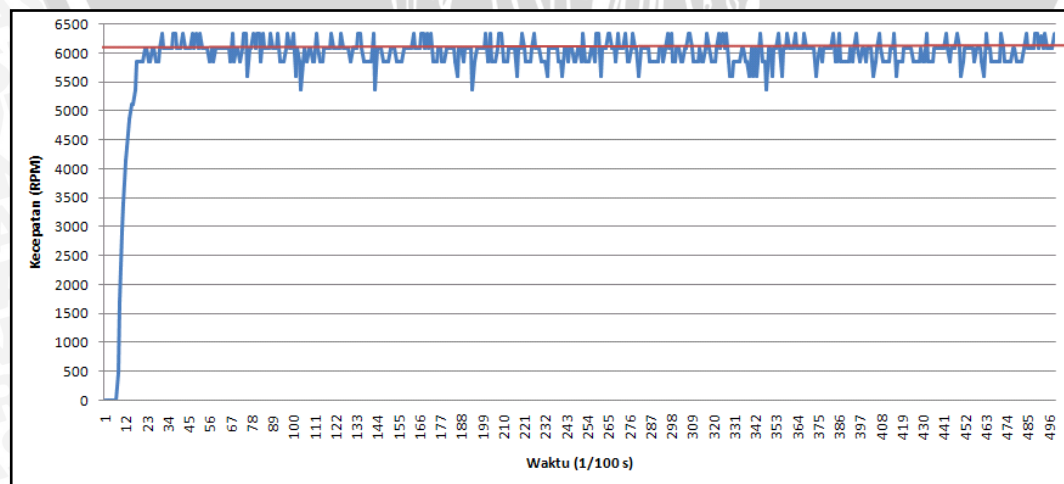
Untuk nilai $K_p = 1.05$, $K_i = 0.035$, dan $K_d = 6.5$ diperoleh grafik respon seperti pada Gambar 5.28.



Gambar 5.28 Grafik Respon Motor DC dengan nilai $K_p=1.05$, $K_i=0.035$, dan $K_d=6.5$

$$\begin{aligned} \%Ess &= \frac{|Averagespeedsteady - setpoint|}{setpoint} \times 100\% \\ &= \frac{|6059.699 - 6098|}{6098} \times 100\% \\ &= 0.63\% \text{ dengan waktu } steady \text{ 0.21 detik.} \end{aligned}$$

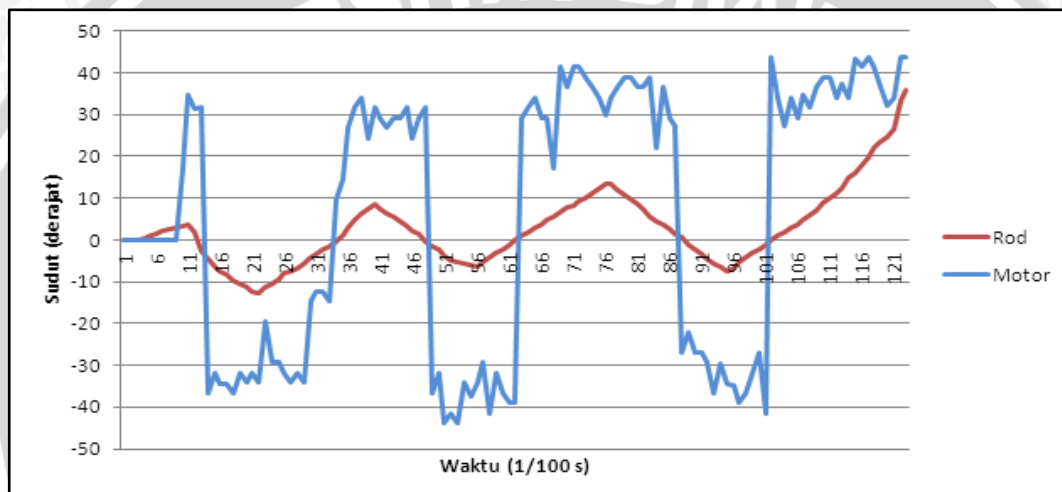
Untuk nilai $K_p = 1.05$, $K_i = 0.04$, dan $K_d = 6.5$ diperoleh grafik respon seperti pada Gambar 5.29.



Gambar 5.29 Grafik Respon Motor DC dengan nilai $K_p=1.05$, $K_i=0.04$, dan $K_d=6.5$

$$\begin{aligned} \%Ess &= \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100\% \\ &= \frac{|6038.990 - 6098|}{6098} \times 100\% \\ &= 0.97\% \text{ dengan waktu } steady \text{ 0.23 detik.} \end{aligned}$$

Dari hasil *hand tuning* diperoleh nilai parameter PID, yaitu $K_p=1.05$, $K_i=0.03$, dan $K_d=6.5$ didapatkan respon yang baik, dan hanya memiliki *error steady state* sebesar 0.43 % dengan waktu *steady* 0.20 detik. Dengan nilai K_p , K_i , K_d yang sudah diperoleh, grafik respon *Rotary Inverted Pendulum* ditunjukkan pada Gambar 5.30.



Gambar 5.30 Grafik Respon *Rotary Inverted Pendulum* dengan nilai $K_p=1.05$, $K_i=0.03$, dan $K_d=6.5$

Berdasarkan pengujian yang telah dilakukan dengan nilai PID, $K_p=1.05$, $K_i=0.03$, dan $K_d=6.5$ menunjukkan bahwa *Rotary Inverted Pendulum* mempunyai respon yang lebih baik dari sebelumnya. *Rotary Inverted Pendulum* dapat mempertahankan posisinya selama 2 detik.

Sistem *Rotary Inverted Pendulum* dapat mengambil keputusan untuk pengaturan arah dan kecepatan sebagai upaya dalam menstabilkan lengan pendulum.