BAB V

PENGUJIAN DAN ANALISIS

Tujuan pengujian dan analisis ini adalah untuk menentukan dan menguji apakah alat yang telah dibuat dapat berfungsi dengan baik sesuai dengan apa yang dikehendaki. Pengujian pada sistem meliputi pengujian setiap blok maupun pengujian secara keseluruhan. Pengujian setiap blok dilakukan agar dapat mempermudah analisis dan menemukan letak kesalahan jika sistem tidak bekerja sesuai dengan perancangan. Pengujian dibagi menjadi beberapa bagian, yaitu :

1. Pengujian sensor tekanan MPX53DP dengan pembanding alat ukur berupa meteran
2. Pengujian *driver* motor DC
3. Pengujian kecepatan hisap dan pompa piston *ballast*
4. Pengujian seluruh sisten tanpa gangguan
5. Pengujian sistem dengan gangguan

5.1 Pengujian Sensor Tekanan dengan meteran

Dalam pengujian sensor tekanan MPX53DP terdapat beberapa aspek yang harus diperhatikan, antara lain :

5.1.1 Tujuan

Mengetahui jarak yang terbaca pada sensor tekanan MPX53 dan membandingkannya dengan alat ukur lain yaitu meteran

5.1.2 Peralatan yang digunakan

1. Pipa ukuran 2 meter
2. Meter ukur
3. Arduino Mega 2560

5.1.3 Langkah pengujian

1. Sensor tekanan MPX53DP diletakan pada ujung pipa bagian bawah
2. Pipa diisi air setiap 10 cc
3. Pembacaan sensor setiap 10 cm dibandingkan dengan alat ukur lain berupa meteran

5.1.4 Hasil Pengujian

Setelah melakukan prosedur pengujian didapatkan data hasil pengujian ditunjukan pada Table 5.1

Tabel 5. 1 Hasil Pengujian Sensor Tekanan MPX53DP

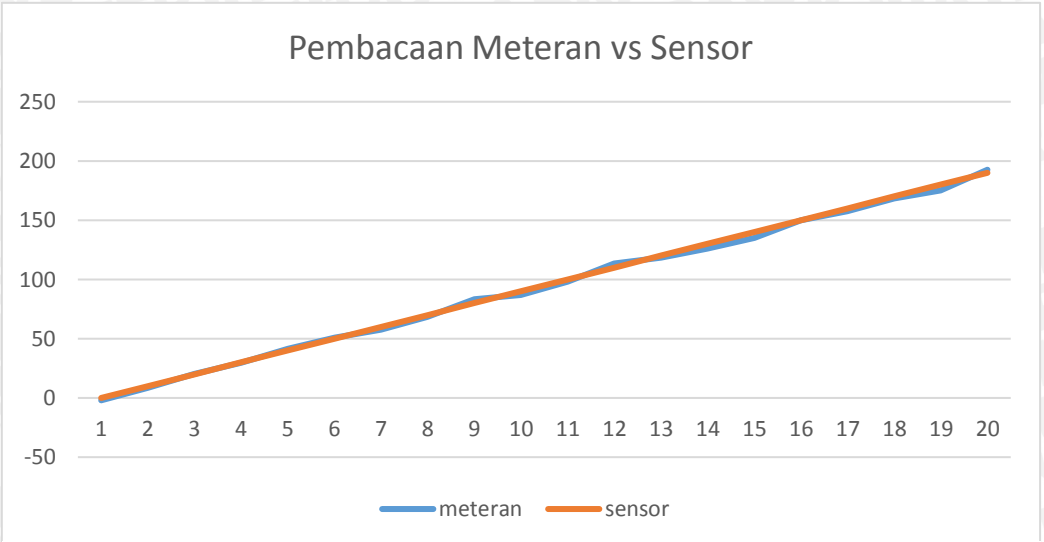
No	Meteran(cm)	Sensor (cm)	No	Meteran (cm)	Sensor (cm)
1	0	-2	11	100	98
2	10	8.7	12	110	113.6
3	20	20.23	13	120	118.5
4	30	29.6	14	130	126
5	40	41.5	15	140	135
6	50	50.79	16	150	149.8
7	60	57.5	17	160	157.7
8	70	68.5	18	170	168.2
9	80	83.2	19	180	175
10	90	87	20	190	192.6

Perhitungan kesalahan pembacaan sensor:

$$\%\ kesalahan = \frac{(\sum Hasil\ Pengukuran - \sum Referensi)}{\sum Referensi} \times 100\%$$

$$\%\ kesalahan = \frac{(1879{,}42 - 1900)}{1900} \times 100\% = 1{,}08\%$$

Tabel 5.1 memperlihatkan bahwa nilai pembacaan yang dihasilkan oleh sensor tekanan udara MPX53DP sudah sangat baik dan sangat mendekati nilai pada meteran dengan nilai persen kesalahan sebesar 1,08%. Berikut perbandingan pembacaan sensor dan meteran dalam bentuk grafik pada Gambar 5.1.



***Gambar* 5. 1** *Grafik perbandingan sensor tekanan dengan meteran*

5.2 Pengujian *Driver* motor DC

5.2.1 Tujuan

Mengetahui output driver motor apabila diberi input yang berbeda-beda

5.2.2 Peralatan yang digunakan

1. Arduino Mega 2560
2. *Driver* motor L293N
3. Motor dc
4. Catu daya adaptor 24V
5. Volt meter

5.2.3 Langkah Pengujian

1. Merangkai peralatan
2. Mengisi mikrokontroler Arduino Mega 2560 dengan program yang mengeluarkan *output* pada pin digital
3. Mengaktifkan catu daya 24v untuk *driver* motor DC
4. Mencatat pergerakan motor DC

5.2.4 Hasil Pengujian

Tabel 5. 2 Pengujian *Driver* Motor DC

	Input			**Output**
No	In3	In4	Enable B	Arah Putaran Motor DC
1	X	X	0	Pengereman atau Diam
2	0	0	1	Pengereman atau Diam
3	0	1	1	Motor Berputar Searah Jarum Jam
4	1	0	1	Motor Berputar Berlawanan Arah Jarum Jam
5	1	1	1	Pengereman atau Diam

Berdasarkan data hasil pengujian pada tabel 5.2 diketahui bahwa motor dc akan berputar searah jarum jam (clockwise) ketika In3 *low* dan in4 *high*, dan apabila In3 *high* dan in4 *low* maka motor akan berputar berlawanan arah jarum jam (*counter clockwise*). Dengan demikian dapat disimpulkan bahwa rangkaian *driver* motor dc ini dapat berjalan dengan baik pada sistem yang direncanakan.

5.3 Pengujian kecepatan hisap piston *ballast*

Dalam pengujian kecepatan hisap piston *ballast* terdapat beberapa aspek yang harus diperhatikan, antara lain :

5.3.1 Tujuan

Untuk mengetahui apakah *driver* motor dc dan piston *ballast* dapat berjalan sesuai dengan yang diinginkan

5.3.2 Peralatan yang digunakan

1. Arduino Mega 2560

2. Multimeter
3. *Stop Watch*
4. Arduino IDE

5.3.3 Langkah Pengujian

Mengukur kecepatan hisap dan pompa piston tangki *ballast* dengan nilai *Pulse Width Modulation* (PWM) yang berbeda dengan menggunakan *stopwatch timer*, sehingga didapatkan kecepatan yang diinginkan lalu mengukur besar tegangan yang keluar dari driver

5.3.4 Hasil Pengujian

Hasil pengujian tegangan keluaran *driver*, kecepatan hisap dan pompa piston *ballast* dengan nilai *Pulse Width Modulation* (PWM) yang ditentukan ditunjukkan pada tabel 5.3

***Tabel* 5. 3** perhitungan tegangan keluaran pwm dan kecepatan hisap piston ballast

No	PWM	Output (volt)	Percobaan (120ml)			Kecepatan Hisap(ml/s)			Rata-rata
			t(s) ke-1	t(s) ke-2	t(s)ke-3	1	2	3	(ml/s)
1	0	0	0	0	0	0	0	0	0
2	25	3.8	0	0	0	0	0	0	0
3	50	6.18	0	0	0	0	0	0	0
4	75	8.52	0	0	0	0	0	0	0
5	100	10.86	0	0	0	0	0	0	0
6	125	13.21	0	0	0	0	0	0	0
7	150	15.56	11.4	10.9	11.3	10.53	11.01	10.61	10.71
8	175	17.9	9.5	9.8	9.9	12.63	12.24	12.12	12.33
9	200	20.32	8.9	8.7	8.8	13.48	13.79	13.63	13.63
10	225	22.4	7.8	7.7	7.6	15.38	15.58	15.78	15.58
11	250	23.9	6.6	6.5	6.4	18.18	18.46	18.75	18.46

Dari Table 5.3 keluaran *output driver* motor berubah sesuai dengan perubahan nilai PWM yang diberikan. *Input* tegangan yang dicatu pada *driver* motor adalah 24 Volt. Sehingga *driver* motor dapat digunakan. Sedangkan untuk piston *ballast* baru bisa mulai bekerja pada rentang PWM 150 sampai dengan 250.

5.4 Pengujian keseluruhan sistem

Pengujian ini dilakukan untuk mengetahui seberapa besar nilai kontroler yang dibutuhkan agar sistem bekerja sesuai dengan setpoint 80cm

5.4.1 Peralatan yang digunakan

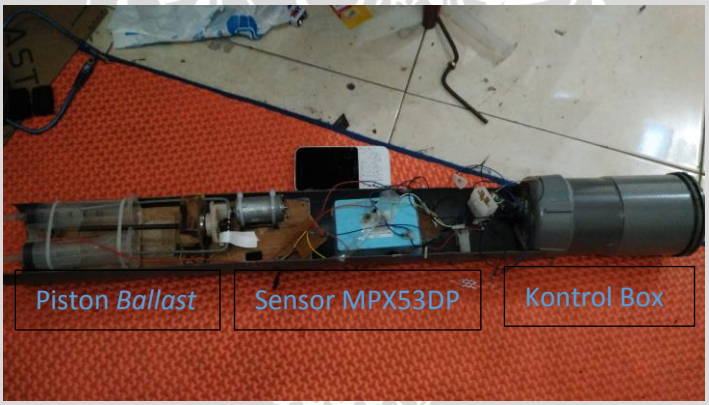
1. Plant tangki *ballast* ROV underwater

2. Sensor tekanan MPX53DP
3. Motor DC
4. Arduino Mega 2560
5. Adapter
6. Meter ukur
7. Data *Logger*

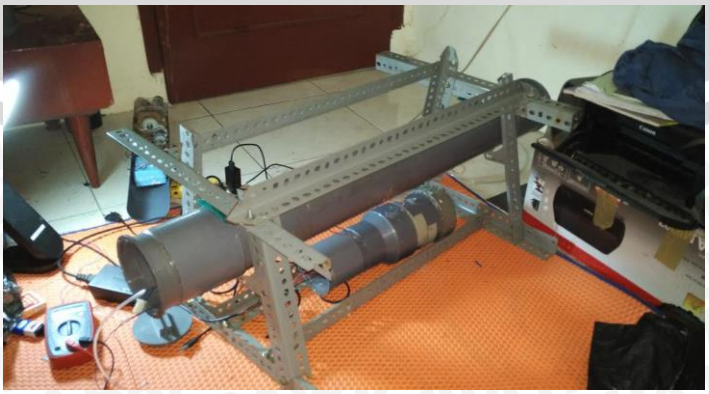
5.4.2 Prosedur Pengujian

1. Menghubungkan *power supply* sebagai sumber tegangan untuk *driver* motor
2. Mengunduh program dengan kontroler PID sesuai parameter yang telah didapat melalui *software* Arduino 1.6.5
3. Setelah memastikan semua rangkaian terpasang dengan benar program dijalankan
4. Pada pengujian pertama kinerja sistem tidak diberi gangguan
5. Membuat grafik dari data yang didapat dari hasil pengujian

   Gambar 5.2 dan Gambar 5.3 adalah gambar keseluruhan sistem yang akan di uji.



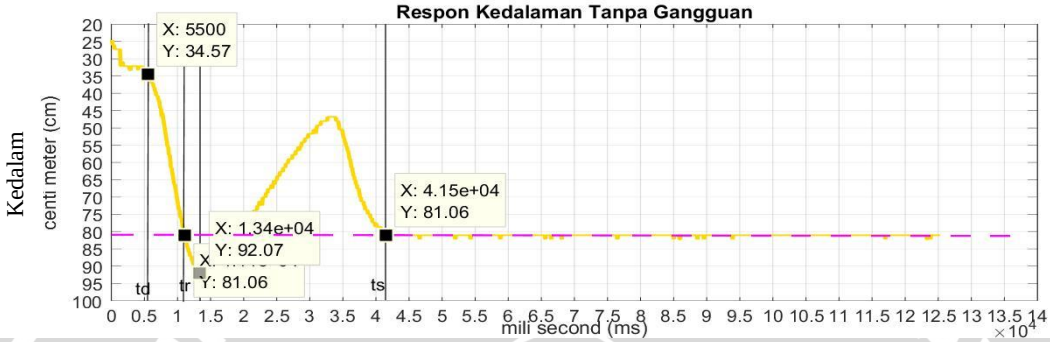
Gambar 5. 2 Piston *ballast*, Sensor, dan kontrol box



***Gambar* 5. 3** *Cassing* pelindung sistem keseluruhan

5.4.3 Hasil Pengujian

Pengujian untuk pengendalian kedalaman pada tangki *ballast* dengan menggunakan kontroler PID dengan Kp= 2,89 Ki=0,74 dan Kd=2,81 dan *setpoint* kedalaman 80 cm didapatkan hasil respon yang ditunjukkan pada Gambar 5.4.



Gambar 5. 4 Respon sistem tanpa gangguan

Hasil dari respon sistem pada Gambar 5.4 memperlihatkan bahwa sistem dapat mencapai kondisi *steady*. Sebelum mencapai kondisi *steady* dan setelah mencapai titik tertinggi kemampuan sistem tedapat kondisi ketika ROV hampir kembali ke posisi awal. Hal itu terjadi dikarenakan terdapat kejenuhan dari piston *ballast*. Piston ballas pada saat bergerak melihat dari hasil pengamatan oleh mata, terkadang terhambat pada saat air mulai menyemburkan di volume 30-40 ml, hal tersebut mengakibatkan air yang keluar terdorong lebih cepat setelah mengalami perlambatan. Dari grafik hasil pengujian untuk pengendalian kedalaman pada tangki *ballast* dengan kedalaman awal 26 cm dengan menggunakan kontroler PID dengan parameter Kp= 2,89 Ki=0,74 dan Kd=2,81 diperoleh data pada Tabel 5.4 sebagai berikut:

Tabel 5. 4 Tabel hasil percobaan sistem tanpa gangguan

Waktu Tunda (td)	5500 mili detik
Waktu Naik (tr)	11000 mili detik
Waktu Puncak (tp)	13400 mili detik
Waktu Keadaan Mantap (ts)	41500 mili detik
Maksimum *Overshoot* (Mp)	13,8 %
Error Steady State	1.3 %

Keterangan :

- Waktu Tunda (t_d) adalah waktu yang diperlukan respons untuk mencapai setengah nilai akhir untuk waktu yang pertama
- Waktu Naik (t_r) adalah aktu yang diperlukan respons untuk naik dari 0-100% dari nilai akhirnya.
- Waktu Puncak (t_p) adalah waktu yang diperlukan respon untuk mencapai nilai puncak respon.
- *Error Steady State* adalah selisih antara *setpoint* yang diinginkan dengan kondisi sebenarnya. Perhitungannya adalah sebagai berikut

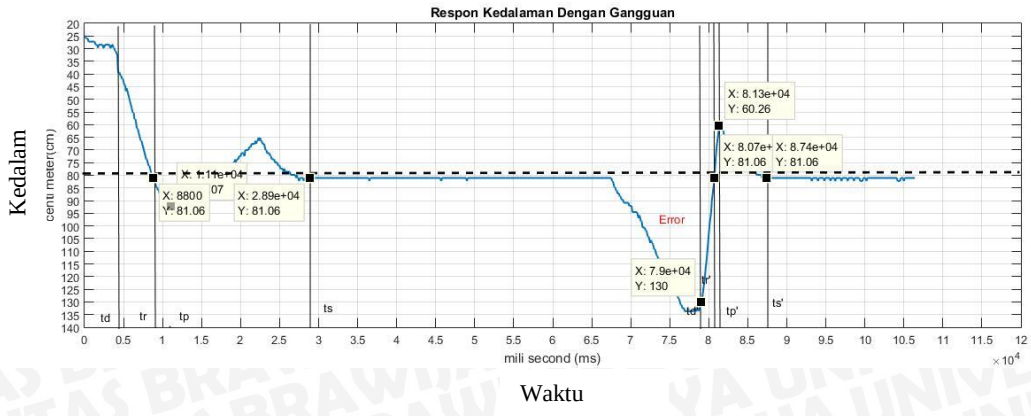
  ((Setpoint-kondisi sebenarnya)/Setpoint)).100%= *Error Steady State* %
- Waktu Keadaan Mantap (t_s) adalah waktu yang dibutuhkan respons untuk mencapai keadaan *steady state.*

Maksimum Puncak (Mp) adalah *overshoot* maksimum yang terjadi saat waktu puncak (t_p). sedangkan untuk presentase dari nilai maksimum *overshoot* adalah sebagai berikut

((Setpoint-titik tertinggi)/Setpoint)).100%= Maksimum puncak %

Pada pengujian sistem pengendalian kedalaman pada tangki *ballast* dengan gangguan berupa dorongan kearah dasar kolam dengan bantuan alat dorong berupa bambu. Pemberian gangguan bertujuan untuk mengetahui apakah sistem dapat mempertahankan posisi kedalaman .

Pengujian sistem pengontrol kedalaman dilakukan dengan menggunakan kontroler PID dengan Kp= 2,89 Ki=0,74 dan Kd=2,81 dan *setpoint* 80 cm dengan kedalaman awal 26,26 cm didapatkan hasil respon yang ditunjukkan pada Gambar 5.5.



Gambar 5. 5 Respon sistem dengan gangguan

Tabel 5. 5 Data perolehan hasil sistem dengan gangguan

Waktu Tunda (td)	1700 mili second
Waktu Naik (tr)	3400 mili second
Waktu Puncak (tp)	4000 mili second
Waktu Keadaan Mantap (ts)	9100 mili second
Maksimum *Overshoot* (Mp)	24,6 %
Error Steady State	3,25 %

Keterangan:

- Waktu Tunda (t_d) adalah waktu yang diperlukan respons untuk mencapai setengah nilai akhir untuk waktu yang pertama.
- Waktu Naik (t_r) adalah aktu yang diperlukan respons untuk naik dari 0-100% dari nilai akhirnya.
- Waktu Puncak (t_p) adalah waktu yang diperlukan respon untuk mencapai nilai puncak respon.
- Waktu Keadaan Mantap (t_s) adalah waktu yang dibutuhkan respons untuk mencapai keadaan *steady state*.
- Maksimum Puncak (Mp) adalah *overshoot* maksimum yang terjadi saat waktu puncak (t_p).