

BAB IV

PENGUJIAN DAN EVALUASI SISTEM

Pada Bab ini membahas tentang pengujian alat berdasarkan sistem yang telah dirancang. Tahap pengujian yang dilakukan mulai dari perangkat keras sampai ke perangkat lunak. Pada pengujian ini akan dilakukan pada perangkat elektronik yang terdapat pada alat pemberi pakan ikan secara otomatis, dimana percobaan ini dikatakan berhasil apabila alat dapat memberikan pakan ikan secara otomatis dan tersebar secara merata pada kolam dengan takaran sesuai yang diinginkan.

4.1 Pengujian *Arduino Mega 2560*

4.1.1 Tujuan

Pengujian *Arduino Mega 2560* bertujuan mengetahui apakah bisa tersambung dengan baik antara mikrokontroler *Arduino Mega 2560* dengan *Software Arduino IDE*. Proses pengujian ini dapat diketahui dengan melakukan proses *compile* program dan *upload* program pada *Software Arduino IDE*.

4.1.2 Alat yang digunakan

Peralatan yang digunakan untuk pengujian ini adalah sebagai berikut :

1. *Microcontroller Arduino Mega 2560*.

2. Kabel *Downloader*.
3. Laptop.
4. *Software Arduino IDE*.
5. *Power Supply 12 V 5 A*

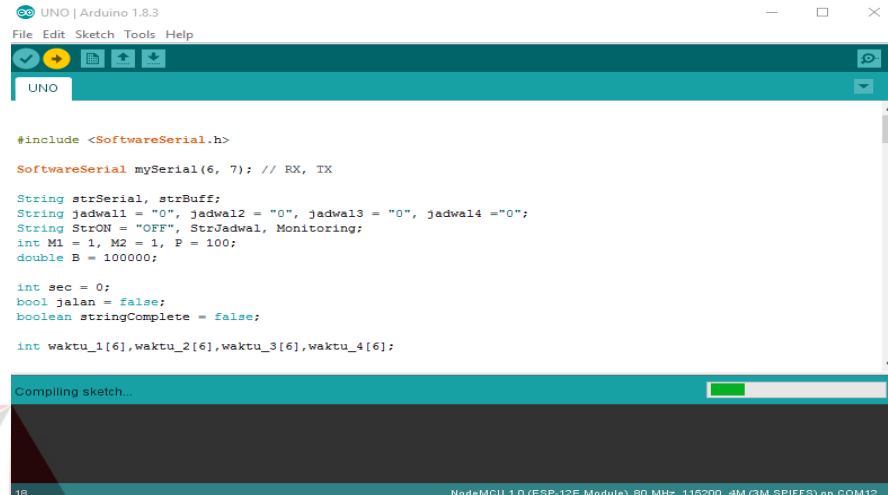
4.1.3 Prosedur Pengujian

1. Mengaktifkan *adaptor power supply 12 V 5 A* dengan *Microcontroller Arduino Mega 2560*.
2. Menghubungkan *Arduino Mega 2560* dengan dengan kabel *downloader* ke laptop.
3. Membuka *Software* program *Arduino IDE*.
4. Membuka *file* program yang akan digunakan lalu *compile* program untuk mengetahui adanya *syntax error* pada program.
5. Setelah tidak ada *syntax error* maka *upload* program yang telah di *compile* ke *Arduino Mega 2560* lalu menunggu hingga proses *upload* program selesai.

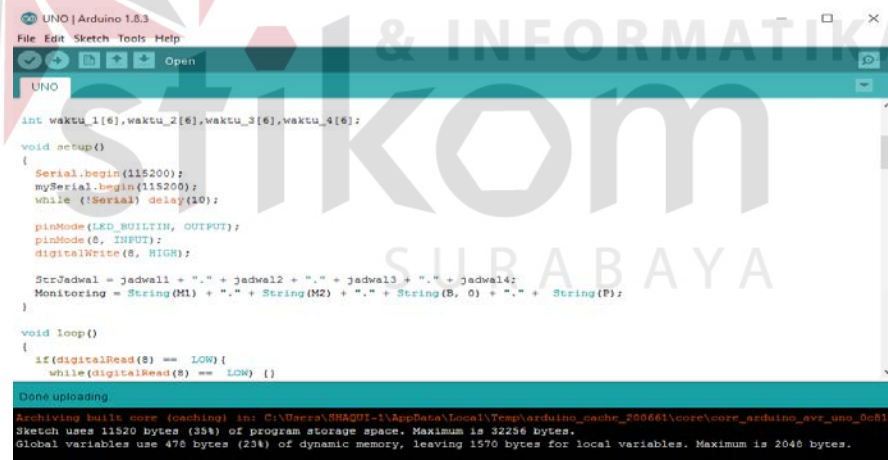
4.1.4 Hasil Pengujian

Dari hasil pengujian saat melakukan *compile* program dan tidak ada *syntax error* maka program dapat digunakan. Untuk proses selanjutnya yaitu

pengujian *upload* program, jika sudah ada tulisan *done uploading* maka program telah berhasil di *upload* di *Microcontroller Arduino Mega 2560*.



Gambar 4.1 Tampilan proses *upload* pada Arduino IDE



Gambar 4.2 Tampilan saat proses *upload* selesai

4.2 Pengujian LCD (*Liquid Crystal Display*)

LCD (*Liquid Crystal Display*) digunakan untuk menampilkan menu dan proses input data pada alat pemberi pakan ikan otomatis. Terdapat empat menu yang akan ditampilkan pada LCD (*Liquid Crystal Display*) yaitu :

1. Berat pakan
2. Penjadwalan
3. Takaran Pakan
4. Start

4.2.1 Tujuan

Tujuan Pengujian LCD (*Liquid Crystal Display*) untuk mengetahui apakah LCD (*Liquid Crystal Display*) sudah dapat terhubung dengan *Arduino Mega 2560* dan berjalan sesuai dengan rancangan penempatan karakter pada LCD (*Liquid Crystal Display*).

4.2.2 Alat yang digunakan

Peralatan yang digunakan untuk pengujian ini adalah sebagai berikut :

1. *Microcontroller Arduino Mega 2560*.
2. Laptop.
3. *Power supply 12 V*.
4. LCD (*Liquid Crystal Display*).
5. *Variable Resistor*.

6. Kabel *jumper*

4.2.3 Prosedur pengujian

1. Menghubungkan *Arduino Mega* 2560 dengan laptop.
2. Menghubungkan LCD (*Liquid Crystal Display*) dengan *Arduino Mega* 2560 menggunakan kabel *jumper*.
3. Menghubungkan LCD (*Liquid Crystal Display*) dengan *Variable* Resistor.
4. Memastikan program telah di *upload*.

4.2.4 Hasil Pengujian

Dari LCD (*Liquid Crystal Display*) menunjukkan hasil tampilan yang sesuai dengan program yang telah dibuat dan di *upload* pada *Arduino Mega* 2560, maka penempatan karakter sudah sesuai dengan yang diinginkan. LCD (*Liquid Crystal Display*) digunakan untuk menampilkan masukan dari *user* berupa *inputan* berat pakan, jadwal pakan, dan takaran pakan serta menampilkan proses ketika alat telah dijalankan.



Gambar 4.3 Tampilan menu pada LCD (*Liquid Crystal Display*)

4.3 Pengujian Keypad

Pada penelitian ini *keypad* digunakan untuk memberikan nilai *input* pada alat dan akan ditampilkan melalui LCD (*Liquid Crystal Display*). Selanjutnya *Inputan* akan diolah dan diproses oleh *Mikrokontroler Arduino Mega 2560*.

4.3.1 Tujuan

Tujuan pengujian ini adalah untuk mengetahui apakah karakter yang telah ditekan pada tombol *keypad* yang digunakan sudah sesuai dengan angka dan huruf yang tertera pada tombol *keypad*.

4.3.2 Alat yang digunakan

Peralatan yang digunakan untuk pengujian *keypad* adalah sebagai berikut :

1. Laptop.
2. LCD (*Liquid Crystal Display*).

3. *Variable Resistor*
4. *Power Supply 12 V*
5. *Keypad Matriks 4 x 4.*
6. *Kabel Jumper.*

4.3.3 Prosedur Pengujian

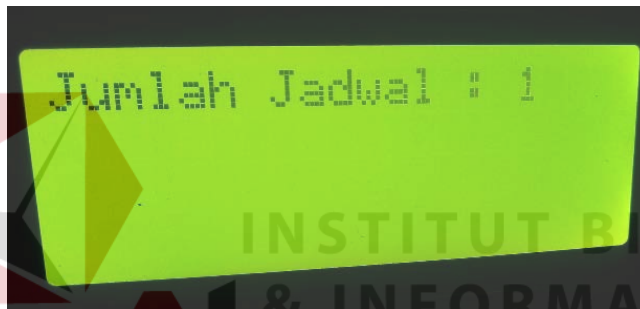
1. Menghubungkan *Arduino Mega 2560* dengan Laptop.
2. Mengaktifkan *Power Supply 12 V*.
3. Menghubungkan LCD (*Liquid Crystal Display*) dengan *Variable Resistor*.
4. Menghubungkan LCD (*Liquid Crystal Display*) dengan *Arduino Mega 2560*.
5. Menghubungkan *Keypad* dengan *Arduino Mega 2560*.

4.3.4 Hasil Pengujian

Hasil pengujian pada bagian ini adalah ketika *user menginputkan data* berdasarkan tombol *keypad* yang ditekan dan ditampilkan pada LCD (*Liquid Crystal Display*). Hasil pengujiannya data yang ditampilkan pada LCD (*Liquid Crystal Display*) sesuai dengan tombol yang ditekan pada *keypad*.



Gambar 4.4 Pengujian penekanan tombol *keypad*



Gambar 4.5 Pengujian penekanan tombol *keypad* pada LCD (*Liquid Crystal Display*)

Tabel 4.1 Hasil pengujian penekanan tombol karakter *keypad* pada LCD (*Liquid Crystal Display*)

No	Penekanan Tombol <i>Keypad</i>	Output Penekanan pada LCD	Status
1	3	3	Akurat
2	2	2	Akurat
3	A	A	Akurat
4	C	C	Akurat
5	5	5	Akurat
6	#	#	Akurat
7	1	1	Akurat
8	B	B	Akurat
9	0	0	Akurat

10	9	9	Akurat
11	3	3	Akurat
12	8	8	Akurat
13	*	*	Akurat
14	A	A	Akurat
15	C	C	Akurat
16	0	0	Akurat
17	1	1	Akurat
18	2	2	Akurat
19	4	4	Akurat
20	6	6	Akurat
21	D	D	Akurat
22	A	A	Akurat
23	9	9	Akurat
24	5	5	Akurat
25	*	*	Akurat
26	#	#	Akurat
27	2	2	Akurat
28	5	5	Akurat
29	0	0	Akurat
30	8	8	Akurat

Pada Tabel 4.1 Hasil pengujian penekanan tombol karakter *keypad* pada LCD (*Liquid Crystal Display*) dilakukan pengujian sebanyak 30 kali dan tidak ditemukan adanya data yang *error*. Jadi disimpulkan persentase keakuratan data pada pengujian ini sebesar 100 %.

4.4 Pengujian Motor Driver dan Motor DC

Pada pengujian ini adalah pengujian terhadap motor *driver* yang berfungsi untuk menjalankan motor DC untuk menggerakkan ulir dan pelontar pakan sesuai program pada *Arduino Mega 2560*.

4.4.1 Tujuan

Tujuan pengujian ini adalah untuk mengetahui apakah motor DC dapat berjalan ketika alat bekerja. Pada pengujian ini motor DC digunakan sebagai sarana penggerak ulir dan pelontar pakan.

4.4.2 Alat yang digunakan

Peralatan yang dibutuhkan untuk pengujian motor *driver* dan motor DC ini adalah sebagai berikut :

1. Laptop.
2. *Mikrokontroller Arduino Mega 2560.*
3. *Power Supply 12 V 5 A.*
4. *Keypad.*
5. *LCD (Liquid Crystal Display).*
6. *Motor Driver EMS 2A Dual H-Bridge.*
7. Motor DC.
8. Kabel *Jumper.*

4.4.3 Prosedur Pengujian

1. Menghubungkan *Arduino Mega 2560* dengan Laptop.
2. Mengaktifkan *Power Supply 12 V.*
3. Menghubungkan LCD (*Liquid Crystal Display*) dengan *Variable Resistor.*

4. Menghubungkan LCD (*Liquid Crystal Display*) dengan *Arduino Mega* 2560.
5. Menghubungkan *Keypad* dengan *Arduino Mega* 2560.
6. Menghubungkan motor *driver* EMS 2A *Dual H-Bridge* dan motor DC ke *Arduino Mega* 2560.

4.4.4 Hasil Pengujian

Pada hasil pengujian ini dilakukan terhadap motor *driver* dan motor DC dimana motor *driver* dapat men-*drive* motor DC untuk menggerakkan ulir dan pelontar pakan. Hasil pengujiannya adalah motor *driver* dapat men-*drive* motor DC untuk menggerakkan ulir dan pelontar pakan. Untuk mengetahui kondisi saat motor aktif dapat dilihat pada gambar 4.6.



Gambar 4.6 Motor *driver* aktif

Tabel 4.2 Hasil pengujian kinerja motor *driver* dan motor DC pada alat

No	Status Keadaan Alat	Status Motor DC	Status
1	ON	ON	Akurat
2	ON	ON	Akurat
3	ON	ON	Akurat
4	ON	ON	Akurat
5	ON	ON	Akurat
6	ON	ON	Akurat
7	ON	ON	Akurat
8	ON	ON	Akurat
9	ON	ON	Akurat
10	ON	ON	Akurat
11	ON	ON	Akurat
12	ON	ON	Akurat
13	ON	ON	Akurat
14	ON	ON	Akurat
15	ON	ON	Akurat
16	ON	ON	Akurat
17	ON	OFF	Tidak Akurat
18	ON	ON	Akurat
19	ON	ON	Akurat
20	ON	ON	Akurat
21	ON	ON	Akurat
22	ON	ON	Akurat
23	ON	ON	Akurat
24	ON	ON	Akurat
25	ON	ON	Akurat
26	ON	ON	Akurat
27	ON	ON	Akurat
28	ON	ON	Akurat
29	ON	ON	Akurat
30	ON	ON	Akurat

Pada Tabel 4.2 Hasil pengujian kinerja motor *driver* dan motor DC Pada Alat dilakukan pengujian sebanyak 30 kali dan hanya ditemukan adanya satu data yang *error*. Jadi disimpulkan persentase keakuratan data pada pengujian ini sebesar 97 %.

4.5 Pengujian RTC (*Real Time Clock*)

Pada penelitian ini RTC (*Real Time Clock*) digunakan sebagai acuan waktu real dengan waktu yang akan diinputkan oleh *user*. Waktu pada RTC di set sama dengan waktu yang sekarang.

4.5.1 Tujuan

Tujuan pengujian ini adalah untuk mengetahui apakah waktu pada RTC (*Real Time Clock*) sesuai dengan waktu yang sekarang. Pada pengujian ini RTC (*Real Time Clock*) digunakan sebagai acuan waktu dalam proses penjadwalan pakan.

4.5.2 Alat yang digunakan

Peralatan yang digunakan untuk pengujian RTC (*Real Time Clock*) adalah sebagai berikut :

1. Laptop.
2. Mikrokontroller Arduino Mega 2560.
3. RTC (*Real Time Clock*) DS3231.
4. LCD (*Liquid Crystal Display*).
5. Variable Resistor
6. Power Supply 12 V
7. Keypad Matriks 4 x 4.

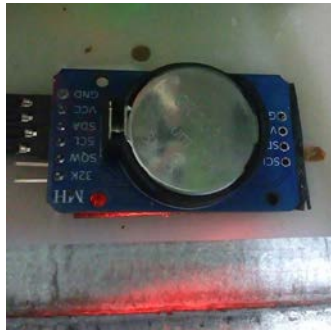
8. Kabel *Jumper*.

4.5.3 Prosedur Pengujian

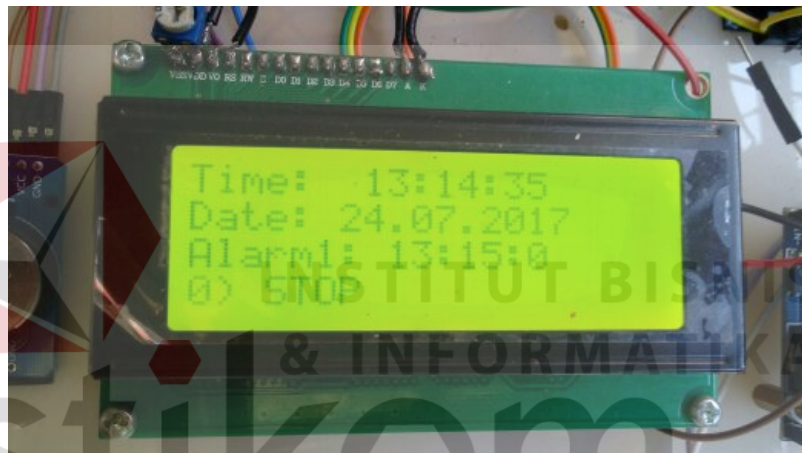
1. Menghubungkan *Arduino Mega* 2560 dengan Laptop.
2. Mengaktifkan *Power Supply* 12 V.
3. Menghubungkan LCD (*Liquid Crystal Display*) dengan *Variable Resistor*.
4. Menghubungkan LCD (*Liquid Crystal Display*) dengan *Arduino Mega* 2560.
5. Menghubungkan *Keypad* dengan *Arduino Mega* 2560.
6. Menghubungkan RTC (*Real Time Clock*) dengan *Arduino Mega* 2560.

4.5.4 Hasil Pengujian

Hasil pengujian pada bagian ini adalah ketika waktu yang di *set* pada RTC (*Real Time Clock*) sesuai dengan waktu *real* dan data waktu RTC (*Real Time Clock*) akan ditampilkan pada LCD (*Liquid Crystal Display*). Hasil pengujiannya data waktu RTC (*Real Time Clock*) yang ditampilkan pada LCD (*Liquid Crystal Display*) sesuai dengan waktu *real*.



Gambar 4.7 RTC (*Real Time Clock*) aktif



Gambar 4.8 Pengujian RTC (*Real Time Clock*) pada LCD
(*Liquid Crystal Display*)

4.6 Pengujian Sensor Rotary

Pada penelitian ini sensor *Rotary* digunakan sebagai acuan untuk menentukan jumlah takaran pakan ikan pada setiap kali penjadwalan pakan dengan jumlah takaran yang akan diinputkan oleh *user*.

4.6.1 Tujuan

Tujuan pengujian ini adalah untuk mengetahui apakah waktu pada sensor *Rotary* dapat membaca putaran pada motor DC. Pada pengujian ini sensor *Rotary* digunakan sebagai acuan untuk menentukan jumlah takaran pakan.

4.6.2 Alat yang digunakan

Peralatan yang digunakan untuk pengujian sensor *Rotary* adalah sebagai berikut :

1. Laptop.
2. *Mikrokontroller Arduino Mega 2560.*
3. *Sensor Rotary.*
4. *Motor DC.*
5. *Motor driver.*
6. *LCD (Liquid Crystal Display).*
7. *Variable Resistor*
8. *Power Supply 12 V*
9. *Kabel Jumper.*

4.6.3 Prosedur Pengujian

1. Menghubungkan *Arduino Mega 2560* dengan Laptop.
2. Mengaktifkan *Power Supply 12 V.*

3. Menghubungkan LCD (*Liquid Crystal Display*) dengan *Variable Resistor*.
4. Menghubungkan LCD (*Liquid Crystal Display*) dengan *Arduino Mega 2560*.
5. Menghubungkan *Keypad* dengan *Arduino Mega 2560*.
6. Menghubungkan *Motor driver* dan motor DC dengan *Arduino Mega 2560*.
7. Menghubungkan motor DC dan sensor *Rotary* dengan *Arduino Mega 2560*.

4.6.4 Hasil Pengujian

Hasil pengujian pada bagian ini adalah ketika motor DC berputar lalu hasil jumlah putaran dapat dibaca oleh sensor *Rotary* dan ditampilkan pada LCD (*Liquid Crystal Display*). Hasil pengujiannya sensor *Rotary* dapat membaca putaran dan ditampilkan pada LCD (*Liquid Crystal Display*).



Gambar 4.9 Sensor *Rotary* saat belum mendeteksi satu putaran



Gambar 4.10 Sensor *Rotary* mendeteksi satu putaran

Tabel 4.3 Hasil pengujian menghitung jumlah putaran pada sensor *Rotary*

No	Jumlah Putaran	Jumlah Putaran Pada Program	Selisih
1	1	3	2
2	2	6	4
3	3	10	7
4	4	14	10
5	5	15	10
6	6	20	14
7	7	23	16
8	8	26	18
9	9	27	18
10	10	34	24
11	1	4	3
12	2	7	5
13	3	9	6
14	4	14	10
15	5	16	11
16	6	19	13
17	7	22	15
18	8	25	17
19	9	28	19
20	10	33	23

21	1	3	2
22	2	8	6
23	3	11	8
24	4	15	11
25	5	17	12
26	6	18	12
27	7	22	15
28	8	24	16
29	9	26	17
30	10	33	23

Pada Tabel 4.3 Hasil pengujian menghitung jumlah putaran pada sensor *Rotary* dilakukan pengujian sebanyak 30 kali. Pada setiap pengujian masih terdapat error dalam pembacaan sensor seperti pada saat user hanya menginputkan 1 putaran tetapi sensor membacanya menjadi 3 putaran atau 4 putaran. Dengan begitu dapat disimpulkan pembacaan 1 kali putaran pada sensor dibuat bernilai 3.

4.7 Pengujian Penjadwalan Pakan

Pada pengujian ini penjadwalan pakan digunakan untuk menjadwalkan pemberian pakan ikan secara otomatis. Data jadwal pakan diambil dari *inputan* dari *keypad* yang dilakukan oleh *user* dan ditampilkan melalui LCD (*Liquid Crystal Display*). Selanjutnya *Inputan* akan diolah dan diproses oleh *Mikrokontroller Arduino Mega 2560* dan dicocokkan dengan waktu RTC (*Real Time Clock*).

4.7.1 Tujuan

Pada pengujian ini bertujuan untuk mengetahui apakah data penjadwalan yang telah diinputkan oleh user sesuai dengan waktu RTC (*Real Time Clock*) atau tidak. Jika sesuai maka alat pemberi pakan ikan ini akan bekerja.

4.7.2 Alat yang digunakan

1. Laptop.
2. *Mikrokontroller Arduino Mega 2560.*
3. *RTC (Real Time Clock).*
4. *LCD (Liquid Crystal Display).*
5. *Variable Resistor*
6. *Power Supply 12 V.*
7. *Kabel Jumper.*

4.7.3 Prosedur Pengujian

1. Menghubungkan *Arduino Mega 2560* pada laptop dengan kabel USB.
2. Menghubungkan *RTC (Real Time Clock)* pada *port Arduino Mega 2560* menggunakan kabel *jumper*.
3. Menghubungkan *Keypad* pada *port Arduino Mega 2560* menggunakan kabel *jumper*.

4. Menghubungkan LCD (*Liquid Crystal Display*) pada port *Arduino Mega* 2560 menggunakan kabel *jumper*.
5. Membuka *software Arduino IDE*.
6. Membuka *file* program yang akan digunakan lalu *compile* program.
7. Setelah proses *compile* selesai dan tidak terdapat *error* maka mengupload program.
8. Menunggu hingga proses *upload* program selesai.
9. Setelah itu akan muncul pilhan menu pada LCD (*Liquid Crystal Display*).
Setelah itu menekan tombol angka dua pada *keypad* untuk memilih menu jadwal.
10. Menginputkan waktu penjadwalan sesuai yang dibutuhkan.
11. Menekan tombol empat pada *keypad* untuk memulai program.
12. Setelah itu *Arduino* akan membandingkan waktu RTC (*Real Time Clock*) dengan waktu yang telah diinputkan oleh *user*. Jika waktu keduanya menunjukkan waktu yang sama maka alat akan berjalan.

4.7.4 Hasil Pengujian

Hasil pengujian jadwal pada Alat Pemberi Pakan Ikan Otomatis adalah seperti pada Tabel 4.4 yang dilakukan sebanyak 30 kali.

Tabel 4.4 Hasil pengujian penjadwalan pada
alat pemberi pakan ikan otomatis

No	Keterangan	Input Jadwal Oleh User	Waktu RTC (<i>Real Time Clock</i>).	Status
1.	Jadwal 1	12:09:00	12:09:00	Akurat
	Jadwal 2	00:00:00	00:00:00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
2.	Jadwal 1	09:00:00	09:00:00	Akurat
	Jadwal 2	13.30.00	13.30.00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
3.	Jadwal 1	07:45:00	07:45:00	Akurat
	Jadwal 2	12:49:00	12:49:00	
	Jadwal 3	19.21:00	19.21:00	
	Jadwal 4	00:00:00	00:00:00	
4.	Jadwal 1	06:45:00	06:45:00	Akurat
	Jadwal 2	11:50:00	11:50:00	
	Jadwal 3	15:37:00	15:37:00	
	Jadwal 4	20:21:00	20:21:00	
5.	Jadwal 1	08:35:00	08:35:00	Akurat
	Jadwal 2	00:00:00	00:00:00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
6.	Jadwal 1	05.25:00	05.25:00	Akurat
	Jadwal 2	12:00:00	12:00:00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
7.	Jadwal 1	05:35:00	05:35:00	Akurat
	Jadwal 2	12:12:00	12:12:00	
	Jadwal 3	18:00:00	18:00:00	
	Jadwal 4	00:00:00	00:00:00	
8.	Jadwal 1	05:14:00	05:14:00	Akurat
	Jadwal 2	11:55:00	11:55:00	
	Jadwal 3	17:08:00	17:08:00	
	Jadwal 4	22:45:00	22:45:00	
9.	Jadwal 1	04:50:00	04:50:00	Akurat
	Jadwal 2	00:00:00	00:00:00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
10.	Jadwal 1	06:01:00	06:01:00	Akurat
	Jadwal 2	12:09:00	12:09:00	

	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
11.	Jadwal 1	08:00:00	08:00:00	Akurat
	Jadwal 2	11:34:00	11:34:00	
	Jadwal 3	16:21:00	16:21:00	
	Jadwal 4	00:00:00	00:00:00	
12.	Jadwal 1	04:30:00	04:30:00	Akurat
	Jadwal 2	10:10:00	10:10:00	
	Jadwal 3	14:50:00	14:50:00	
	Jadwal 4	18:55:00	18:55:00	
13.	Jadwal 1	07:09:00	07:09:00	Akurat
	Jadwal 2	00:00:00	00:00:00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
14.	Jadwal 1	06:45:00	06:45:00	Akurat
	Jadwal 2	12:00:00	12:00:00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
15.	Jadwal 1	05:16:00	05:16:00	Akurat
	Jadwal 3	11:54:00	11:54:00	
	Jadwal 3	16:25:00	16:25:00	
	Jadwal 4	00:00:00	00:00:00	
16.	Jadwal 1	06:02:00	06:02:00	Akurat
	Jadwal 2	10:59:00	10:59:00	
	Jadwal 3	13:11:00	13:11:00	
	Jadwal 4	18:00:00	18:00:00	
17.	Jadwal 1	04:40:00	04:40:00	Akurat
	Jadwal 2	00:00:00	00:00:00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
18.	Jadwal 1	05:34:00	05:34:00	Akurat
	Jadwal 2	12:49:00	12:49:00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
19.	Jadwal 1	05:50:00	05:50:00	Akurat
	Jadwal 2	11:01:00	11:01:00	
	Jadwal 3	16:36:00	16:36:00	
	Jadwal 4	00:00:00	00:00:00	
20.	Jadwal 1	05:40:00	05:40:00	Akurat
	Jadwal 2	09:45:00	09:45:00	
	Jadwal 3	12:21:00	12:21:00	
	Jadwal 4	17:00:00	17:00:00	
21.	Jadwal 1	05:00:00	05:00:00	

	Jadwal 2	00:00:00	00:00:00	Akurat
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
22	Jadwal 1	07:00:00	07:00:00	Akurat
	Jadwal 2	12:09:00	12:09:00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
23	Jadwal 1	06:00:00	06:00:00	Akurat
	Jadwal 2	12:34:00	12:34:00	
	Jadwal 3	14:40:00	14:40:00	
	Jadwal 4	00:00:00	00:00:00	
24	Jadwal 1	04:00:00	04:00:00	Akurat
	Jadwal 2	08:00:00	08:00:00	
	Jadwal 3	13:00:00	13:00:00	
	Jadwal 4	17:00:00	17:00:00	
25	Jadwal 1	08:56:00	08:56:00	Akurat
	Jadwal 2	00:00:00	00:00:00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
26	Jadwal 1	05:15:00	05:15:00	Akurat
	Jadwal 2	12:45:00	12:45:00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
27	Jadwal 1	05:02:00	05:02:00	Akurat
	Jadwal 2	09:55:00	09:55:00	
	Jadwal 3	13:09:00	13:09:00	
	Jadwal 4	00:00:00	00:00:00	
28	Jadwal 1	06:00:00	06:00:00	Akurat
	Jadwal 2	11:00:00	11:00:00	
	Jadwal 3	15:09:00	15:09:00	
	Jadwal 4	20:21:00	20:21:00	
29	Jadwal 1	06:17:00	06:17:00	Akurat
	Jadwal 2	00:00:00	00:00:00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
30	Jadwal 1	05:43:00	05:43:00	Akurat
	Jadwal 2	16:49:00	16:49:00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	

Pada Tabel 4.4 pengujian penjadwalan dari alat pemberi pakan ikan otomatis dengan melakukan penjadwalan sebanyak 30 kali, tidak terdapat data yang tidak akurat pada data penjadwalan pakan dikarenakan metode ini menggunakan konsep seperti cara kerja alarm sehingga tidak memerlukan adanya koneksi jaringan. Persentase keakuratan data pada pengujian ini adalah 100 %.

4.8 Pengujian Penyimpanan Jadwal Pakan

Pada pengujian ini digunakan untuk menyimpan jadwal pemberian pakan ikan secara otomatis. Data jadwal pakan diambil dari *inputan* dari *keypad* yang dilakukan oleh *user* dan ditampilkan melalui LCD (*Liquid Crystal Display*). Selanjutnya *Inputan* akan disimpan oleh *Mikrokontroller Arduino Mega 2560*.

4.8.1 Tujuan

Pada pengujian ini bertujuan untuk menyimpan data berupa jadwal pemberian pakan ikan yang telah diinputkan oleh user, sehingga ketika listrik mati jadwal pemberian pakan ikan akan tetap tersimpan.

4.8.2 Alat yang digunakan

1. Laptop.
2. *Mikrokontroller Arduino Mega 2560*.
3. *RTC (Real Time Clock)*.

4. LCD (*Liquid Crystal Display*).
5. *Variable* Resistor
6. *Power Supply* 12 V
7. Kabel *Jumper*.

4.8.3 Prosedur Pengujian

1. Menghubungkan *Arduino Mega* 2560 pada laptop dengan kabel USB.
2. Menghubungkan RTC (*Real Time Clock*) pada *port Arduino Mega* 2560 menggunakan kabel *jumper*.
3. Menghubungkan *Keypad* pada *port Arduino Mega* 2560 menggunakan kabel *jumper*.
4. Menghubungkan LCD (*Liquid Crystal Display*) pada *port Arduino Mega* 2560 menggunakan kabel *jumper*.
5. Membuka *software Arduino IDE*.
6. Membuka *file* program yang akan digunakan lalu *compile* program.
7. Setelah proses *compile* selesai dan tidak terdapat *error* maka *mengupload* program.
8. Menunggu hingga proses *upload* program selesai.
9. Setelah itu akan muncul pilhan menu pada LCD (*Liquid Crystal Display*).
Setelah itu menekan tombol angka dua pada *keypad* untuk memilih menu jadwal.
10. *Menginputkan* waktu penjadwalan sesuai yang dibutuhkan.

11. Menekan tombol lima pada *keypad* untuk melihat menu jadwal yang telah *diinputkan*.
12. Setelah itu mencabut kabel power pada alat, lalu menancapkan kembali.
13. Setelah itu mengecek kembali jadwal yang telah *diinputkan* tadi apakah telah tersimpan atau tidak.

4.8.4 Hasil Pengujian

Hasil pengujian penyimpanan jadwal pada Alat Pemberi Pakan Ikan Otomatis adalah seperti pada Tabel 4.5 yang dilakukan sebanyak 30 kali.

Tabel 4.5 Hasil pengujian penyimpanan jadwal pada alat pemberi pakan ikan otomatis

No	Keterangan	Jadwal Sebelum Listrik Padam	Jadwal Sesudah Listrik Padam	Status
1.	Jadwal 1	10:09:00	10:09:00	Akurat
	Jadwal 2	00:00:00	00:00:00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
2.	Jadwal 1	09:10:00	09:10:00	Akurat
	Jadwal 2	14.40.00	14.40.00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
3.	Jadwal 1	08:40:00	08:40:00	Akurat
	Jadwal 2	15:49:00	15:49:00	
	Jadwal 3	19.29:00	19.29:00	
	Jadwal 4	00:00:00	00:00:00	
4.	Jadwal 1	06:45:00	06:45:00	Akurat
	Jadwal 2	11:50:00	11:50:00	
	Jadwal 3	15:37:00	15:37:00	
	Jadwal 4	20:21:00	20:21:00	
5.	Jadwal 1	08:35:00	08:35:00	Akurat
	Jadwal 2	00:00:00	00:00:00	

	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
6.	Jadwal 1	06:25:00	06:25:00	Akurat
	Jadwal 2	13:00:00	13:00:00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
7.	Jadwal 1	07:35:00	07:35:00	Akurat
	Jadwal 2	13:12:00	13:12:00	
	Jadwal 3	18:00:00	18:00:00	
	Jadwal 4	00:00:00	00:00:00	
8.	Jadwal 1	05:34:00	05:34:00	Akurat
	Jadwal 2	11:55:00	11:55:00	
	Jadwal 3	17:08:00	17:08:00	
	Jadwal 4	20:45:00	20:45:00	
9.	Jadwal 1	06:50:00	06:50:00	Akurat
	Jadwal 2	00:00:00	00:00:00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
10.	Jadwal 1	06:01:00	06:01:00	Akurat
	Jadwal 2	11:19:00	11:19:00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
11.	Jadwal 1	09:00:00	09:00:00	Akurat
	Jadwal 2	11:34:00	11:34:00	
	Jadwal 3	16:21:00	16:21:00	
	Jadwal 4	00:00:00	00:00:00	
12.	Jadwal 1	05:30:00	05:30:00	Akurat
	Jadwal 2	10:10:00	10:10:00	
	Jadwal 3	14:50:00	14:50:00	
	Jadwal 4	18:55:00	18:55:00	
13.	Jadwal 1	07:09:00	07:09:00	Akurat
	Jadwal 2	00:00:00	00:00:00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
14.	Jadwal 1	06:45:00	06:45:00	Akurat
	Jadwal 2	12:00:00	12:00:00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
15.	Jadwal 1	05:16:00	05:16:00	Akurat
	Jadwal 3	11:54:00	11:54:00	
	Jadwal 3	16:25:00	16:25:00	
	Jadwal 4	00:00:00	00:00:00	
16.	Jadwal 1	06:02:00	06:02:00	

	Jadwal 2	10:59:00	10:59:00	Akurat
	Jadwal 3	13:11:00	13:11:00	
	Jadwal 4	18:00:00	18:00:00	
17.	Jadwal 1	04:40:00	04:40:00	Akurat
	Jadwal 2	00:00:00	00:00:00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
18.	Jadwal 1	05:34:00	05:34:00	Akurat
	Jadwal 2	12:49:00	12:49:00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
19	Jadwal 1	05:50:00	05:50:00	Akurat
	Jadwal 2	11:01:00	11:01:00	
	Jadwal 3	16:36:00	16:36:00	
	Jadwal 4	00:00:00	00:00:00	
20	Jadwal 1	05:40:00	05:40:00	Akurat
	Jadwal 2	09:45:00	09:45:00	
	Jadwal 3	12:21:00	12:21:00	
	Jadwal 4	17:00:00	17:00:00	
21	Jadwal 1	05:00:00	05:00:00	Akurat
	Jadwal 2	00:00:00	00:00:00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
22	Jadwal 1	07:00:00	07:00:00	Akurat
	Jadwal 2	12:09:00	12:09:00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
23	Jadwal 1	06:00:00	06:00:00	Akurat
	Jadwal 2	12:34:00	12:34:00	
	Jadwal 3	14:40:00	14:40:00	
	Jadwal 4	00:00:00	00:00:00	
24	Jadwal 1	04:00:00	04:00:00	Akurat
	Jadwal 2	08:00:00	08:00:00	
	Jadwal 3	13:00:00	13:00:00	
	Jadwal 4	17:00:00	17:00:00	
25	Jadwal 1	08:56:00	08:56:00	Akurat
	Jadwal 2	00:00:00	00:00:00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
26	Jadwal 1	05:15:00	05:15:00	Akurat
	Jadwal 2	12:45:00	12:45:00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	

27	Jadwal 1	05:02:00	05:02:00	Akurat
	Jadwal 2	09:55:00	09:55:00	
	Jadwal 3	13:09:00	13:09:00	
	Jadwal 4	00:00:00	00:00:00	
28	Jadwal 1	06:00:00	06:00:00	Akurat
	Jadwal 2	11:00:00	11:00:00	
	Jadwal 3	15:09:00	15:09:00	
	Jadwal 4	20:21:00	20:21:00	
29	Jadwal 1	06:17:00	06:17:00	Akurat
	Jadwal 2	00:00:00	00:00:00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	
30	Jadwal 1	05:43:00	05:43:00	Akurat
	Jadwal 2	16:49:00	16:49:00	
	Jadwal 3	00:00:00	00:00:00	
	Jadwal 4	00:00:00	00:00:00	

Pada Tabel 4.5 pengujian penyimpanan jadwal dari alat pemberi pakan ikan otomatis dengan melakukan penyimpanan jadwal sebanyak 30 kali, tidak terdapat data yang tidak akurat pada data penyimpanan jadwal pemberian pakan. Persentase keakuratan data pada pengujian ini adalah 100 %.

4.9 Pengujian Penakaran Pakan

Pada pengujian ini penakaran pakan digunakan untuk menentukan jumlah pemberian takaran pakan ikan pada setiap kali penjadwalan. Data penakaran pakan diambil dari *inputan* dari *keypad* yang dilakukan oleh *user* dan ditampilkan melalui LCD (*Liquid Crystal Display*). *Inputan* jumlah takaran pakan menggunakan satuan berat gram. Selanjutnya *Inputan* akan diolah dan diproses oleh *Mikrokontroller Arduino Mega 2560*.

4.9.1 Tujuan

Pada pengujian ini bertujuan untuk mengetahui apakah data jumlah takaran pakan yang telah diinputkan oleh *user* sesuai dengan takaran pakan ikan yang dikeluarkan oleh alat atau tidak.

4.9.2 Alat yang digunakan

1. Laptop.
2. *Mikrokontroller Arduino Mega 2560.*
3. LCD (*Liquid Crystal Display*).
4. *Variable Resistor.*
5. *Motor driver.*
6. Motor DC.
7. *Sensor Rotary.*
8. *Power Supply 12 V.*
9. Kabel *Jumper.*

4.9.3 Prosedur Pengujian

1. Mengubungkan *Arduino Mega 2560* pada laptop dengan kabel USB.
2. Menghubungkan *Keypad* pada *port Arduino Mega 2560* menggunakan kabel *jumper*.
3. Menghubungkan LCD (*Liquid Crystal Display*) pada *port Arduino Mega 2560* menggunakan kabel *jumper*.

4. Menghubungkan RTC (Real Time Clock) pada *port Arduino Mega 2560* menggunakan kabel *jumper*.
5. Menghubungkan Motor *Driver* dan Motor DC dengan *Arduino Mega 2560* menggunakan kabel *jumper*.
6. Menghubungkan sensor *Rotary* pada Motor DC dengan *Arduino Mega 2560* menggunakan kabel *jumper*.
7. Memasukkan pakan ikan pada tempat penampung pakan.
8. Membuka *software Arduino IDE*.
9. Membuka *file* program yang akan digunakan lalu *compile* program.
10. Setelah proses *compile* program selesai dan tidak terdapat *error* maka langkah selanjutnya adalah *upload* program.
11. Menunggu hingga proses *upload* program selesai.
12. Setelah itu akan muncul pilihan menu pada LCD (*Liquid Crystal Display*).
Lalu menekan tombol angka dua untuk memilih menu penjadwalan pakan.
13. *Menginputkan* waktu penjadwalan pakan sesuai yang dibutuhkan.
14. Setelah itu akan muncul pilihan menu pada LCD (*Liquid Crystal Display*).
Lalu menekan tombol angka tiga untuk memilih menu penakaran pakan.
15. *Menginputkan* berat takaran sesuai yang dibutuhkan.
16. Menekan tombol empat pada *keypad* untuk memulai program.
17. Setelah itu menunggu hingga memasuki waktu penjadwalan pakan.
18. Saat memasuki penjadwalan pakan maka motor DC akan menggerakkan ulir yang berfungsi sebagai media untuk metode penakaran.

19. Setelah penakaran pakan dilakukan, pakan yang dikeluarkan akan ditimbang menggunakan timbangan berat yang menggunakan satuan berat gram.
20. Menghitung keakuratan data penakaran pakan yang diinputan oleh user dengan data pakan yang telah dikeluarkan oleh alat.

4.9.4 Pengujian Takaran Pakan

Hasil pengujian Takaran pakan pada Alat Pemberi Pakan Ikan Otomatis adalah seperti pada Tabel 4.6 yang dilakukan sebanyak 30 kali.

Tabel 4.6 Hasil pengujian jumlah takaran pakan pada alat pemberi pakan ikan otomatis

No	Input Jumlah Takaran Oleh User	Jumlah Takaran Yang Dikeluarkan Oleh Alat	Selisih
1.	40 gram	40 gram	0 gram
2.	40 gram	40 gram	0 gram
3.	40 gram	40 gram	0 gram
4.	40 gram	49 gram	9 gram
5.	40 gram	46 gram	6 gram
6.	40 gram	38 gram	2 gram
7.	40 gram	50 gram	10 gram
8.	40 gram	44 gram	4 gram
9.	40 gram	48 gram	8 gram
10.	40 gram	47 gram	7 gram
11.	80 gram	81 gram	1 gram
12.	80 gram	86 gram	6 gram
13.	80 gram	91 gram	11 gram
14.	80 gram	85 gram	5 gram
15.	80 gram	84 gram	4 gram
16.	80 gram	86 gram	6 gram
17.	80 gram	82 gram	2 gram
18.	80 gram	92 gram	12 gram

19.	80 gram	88 gram	8 gram
20.	80 gram	94 gram	14 gram
21.	120 gram	129 gram	9 gram
22.	120 gram	136 gram	16 gram
23.	120 gram	129 gram	9 gram
24.	120 gram	130 gram	10 gram
25.	120 gram	127 gram	7 gram
26.	120 gram	127 gram	7 gram
27.	120 gram	133 gram	13 gram
28.	120 gram	122 gram	2 gram
29.	120 gram	129 gram	9 gram
30.	120 gram	138 gram	18 gram

Pada Tabel 4.6 pengujian penakaran dari alat pemberi pakan ikan otomatis dengan melakukan penakaran sebanyak 30 kali, Pada penakaran pakan dengan jumlah takaran 40 gr terdapat selisih terdekat seberat 0 gr dan selisih terjauh seberat 10 gr dengan rata-rata selisih seberat 5 gr. Pada penakaran pakan dengan jumlah takaran 80 gr terdapat selisih terdekat seberat 1 gr dan selisih terjauh seberat 14 gr dengan rata-rata selisih seberat 7 gr. Pada penakaran pakan dengan jumlah takaran 120 gr terdapat selisih terdekat seberat 2 gr dan selisih terjauh seberat 18 gr dengan rata-rata selisih seberat 11 gr. Untuk selisih berat takaran seberat kurang dari atau sama dengan 5 gr dianggap bernilai akurat sedangkan jika berat takaran diatas 5 gr maka dianggap tidak akurat. Persentase keakuratan data pada ketiga kriteria input berat takaran pakan adalah 33 %.

4.10 Pengujian Pelontaran Pakan

Pada pengujian ini pelontaran pakan digunakan untuk melontarkan pakan ke area kolam ikan secara merata. *Input* data pelontaran ikan diambil dari *voltage* pada *Power Supply*. Pada *power supply* terdapat pilihan besar *voltage* yaitu 6 V, 9 V, dan 12 V.

4.10.1 Tujuan

Pada pengujian ini bertujuan untuk mengetahui berapakah jarak pelontaran pakan ikan yang dilontarkan melalui pelontar pakan dengan *input* pelontaran pakan berupa *voltage* pada *power supply*. Pelontaran pakan berguna untuk menyebarkan pakan ikan ke area kolam secara merata sehingga tidak menumpuk pada satu titik.

4.10.2 Alat yang digunakan

1. Laptop.
2. Mikrokontroler Arduino Mega 2560.
3. Power Supply 12 V.
4. Motor driver.
5. Motor DC.
6. Pelontar Pakan.
7. Sensor Rotary.
8. Kabel Jumper.
9. Pakan ikan.

4.10.3 Prosedur Pengujian

1. Menghubungkan *Power Supply* 12 V 3 A dengan motor DC yang telah terhubung dengan pelontar pakan menggunakan kabel *jumper*;
2. Menyediakan pakan ikan berukuran 2 mm.
3. Menyalakan *power supply* 12 V 3 A, setelah dalam keadaan ON pelontar pakan akan berputar.
4. Memilih *inputan voltage* sesuai yang dibutuhkan.
5. Menjatuhkan pakan tepat diatas pelontar pakan.
6. Mengukur jarak dari pelontaran pakan tersebut.

4.10.4 Pengujian Pelontaran Pakan

Hasil pengujian Pelontaran pakan pada Alat Pemberi Pakan Ikan Otomatis adalah seperti pada Tabel 4.7 yang dilakukan sebanyak 30 kali. Pada kecepatan 6 V diambil 10 sampel jarak pelontaran pakan, dari 10 sampel tersebut maka dapat dihitung rata-rata jarak pelontaran pakan sejauh 125 cm. Nilai tersebut akan digunakan sebagai acuan jarak pelontaran pakan pada kecepatan 6 V. Pada kecepatan 9 V diambil 10 sampel jarak pelontaran pakan, dari 10 sampel tersebut maka dapat dihitung rata-rata jarak pelontaran pakan sejauh 170 cm. Nilai tersebut akan digunakan sebagai acuan jarak pelontaran pakan pada kecepatan 9 V. Pada kecepatan 12 V diambil 10 sampel jarak pelontaran pakan, dari 10 sampel tersebut maka dapat dihitung rata-rata jarak pelontaran pakan sejauh 211 cm. Nilai tersebut akan digunakan sebagai acuan jarak pelontaran pakan pada kecepatan 12 V.

Tabel 4.7 Hasil pengujian pelontaran pakan pada alat pemberi pakan ikan otomatis.

No	Kecepatan Motor	Jarak Rata-Rata Pelontaran	Jarak Pelontaran	Selisih Jarak
1.	6 V	125 cm	124 cm	1 cm
2.	6 V	125 cm	122 cm	3 cm
3.	6 V	125 cm	130 cm	5 cm
4.	6 V	125 cm	132 cm	7 cm
5.	6 V	125 cm	129 cm	4 cm
6.	6 V	125 cm	135 cm	10 cm
7.	6 V	125 cm	120 cm	5 cm
8.	6 V	125 cm	122 cm	3 cm
9.	6 V	125 cm	129 cm	4 cm
10.	6 V	125 cm	136 cm	11 cm
11.	9 V	170 cm	171 cm	1 cm
12.	9 V	170 cm	167 cm	3 cm
13.	9 V	170 cm	183 cm	7 cm
14.	9 V	170 cm	177 cm	7 cm
15.	9 V	170 cm	173 cm	3 cm
16.	9 V	170 cm	169 cm	1 cm
17.	9 V	170 cm	166 cm	4 cm
18.	9 V	170 cm	172 cm	2 cm
19.	9 V	170 cm	179 cm	9 cm
20.	9 V	170 cm	170 cm	0 cm
21.	12 V	211 cm	205 cm	5 cm
22.	12 V	211 cm	211 cm	0 cm
23.	12 V	211 cm	214 cm	3 cm
24.	12 V	211 cm	209 cm	2 cm
25.	12 V	211 cm	220 cm	1 cm
26.	12 V	211 cm	216 cm	5 cm
27.	12 V	211 cm	213 cm	2 cm
28.	12 V	211 cm	207 cm	4 cm
29.	12 V	211 cm	219 cm	8 cm
30.	12 V	211 cm	210 cm	1 cm

Dari hasil percobaan diatas pada pelontaran pakan dengan *input voltage* sebesar 6 v terdapat selisih jarak terdekat sepanjang 1 cm dan selisih terjauh sepanjang 11 cm dengan rata-rata selisih sepanjang 5 cm. Pada

pelontaran pakan dengan *input voltage* sebesar 9 v terdapat selisih jarak terdekat sepanjang 0 cm dan selisih terjauh sepanjang 9 cm dengan rata-rata selisih sepanjang 4 cm. Pada pelontaran pakan dengan *input voltage* sebesar 12 v terdapat selisih jarak terdekat sepanjang 0 cm dan selisih terjauh sepanjang 8 cm dengan rata-rata selisih sepanjang 3 cm.

Untuk selisih berat takaran seberat kurang dari atau sama dengan 3 gr dianggap bernilai akurat sedangkan jika berat takaran diatas 3 gr maka dianggap tidak akurat. Persentase keakuratan data pada ketiga kriteria input berat takaran pakan adalah 50 %.

