

BAB IV HASIL DAN PEMBAHASAN

Pada bagian ini akan membahas mengenai hasil penelitian dalam pembuatan modul “ Otomatisasi Pemberian Pakan Ikan Berbasis *Internet of Things* (IOT) “ dan pada bagian ini akan membahas mengenai pengujian alatnya, dimana pengujian alat ini bertujuan untuk mengetahui kebenaran rangkaian dan kesesuaian kerja alatnya dengan fitur yang diinginkan. Pengujian alat ini terdiri dari pengujian rangkaian catu daya, pengujian tampilan Oled D1 Mini, pengujian fungsional alat, dan pengujian berat pakan yang dikeluarkan, selain pengujian akan dibahas monitoring level pakan.

4.1 Pengujian Catu Daya

Pengujian catu daya dilakukan untuk mengetahui tegangan yang dihasilkan oleh rangkaian catu daya sudah sesuai dengan yang diharapkan atau tidak menggunakan multimeter digital.

Berikut merupakan tabel 4.1 pengujian rangkaian catu daya :

Percobaan Ke-	Komponen	Status	Tegangan Keluaran		Persentase Kesalahan
			Rekomendasi	Pengukuran	
1	LM 7805	Standby	5V	4,94	1 %
2	LM7805	Servo Aktif	5V	4,84	3 %

Pengamatan dilakukan dengan mengukur tegangan catu daya (*power supply*) menggunakan multimeter digital. Rangkaian catu daya terdiri dari komponen transformator *stepdown*. Transformator ini mendapatkan suplai dari tegangan PLN, kemudian tegangan tersebut diturunkan dari 220V AC pada sisi primer menjadi 15V AC pada sisi sekunder. Tegangan keluaran dari trafo kemudian diserahkan dengan rangkaian diode bridge sehingga menghasilkan tegangan DC yang distabilkan dengan bantuan kapasitor. Untuk mendapatkan 5V DC maka digunakan

IC LM 7805 untuk sumber +5 volt yang selanjutnya distabilkan dengan kapasitor yang bertujuan untuk mengurangi riak dan memperhalus tegangan.

Berdasarkan hasil pengujian *power supply*, besarnya tegangan keluaran IC LM7805 saat status standby adalah 4,94 volt dan pada saat status servo aktif adalah 4,84. Idealnya besar tegangan keluaran IC LM7805 adalah 5 volt. Penyimpangan tegangan keluaran sebesar :

$$Error (\%) = \frac{|5-4,95|}{5} \times 100\% = 1\% \quad (3.3)$$

$$Error (\%) = \frac{|5-4,85|}{5} \times 100\% = 3\% \quad (3.3)$$

Penyimpangan yang terjadi cukup kecil yaitu sebesar 1% dan 3%.

4.2 Pengujian Tampilan Oled D1 Mini

Pengujian pada tampilan Oled D1 Mini dilakukan untuk mengetahui sesuai atau tidaknya saat alarm dan jam disetting melalui aplikasi dengan tampilan pada Oled D1 Mini. Berikut adalah tabel 4.2 pengujiannya :

Tabel 4.2 Hasil Pengujian Tampilan Oled D1 Mini

No.	Setting Alarm dan Jam Pada Aplikasi	Tampilan Oled D1 Mini
1	Set Alarm 1 : 06:00	Alarm 1 : 06:00
2	Set Alarm 2 : 14:00	Alarm 2 : 14:00
3	Set Alarm 3 : 21:00	Alarm 3 : 21:00
4	Set Timer : 23:44:10	23:44:10

Berdasarkan tabel pengujian diatas pada bagian tampilan Oled D1 Mini yang telah dilakukan dapat disimpulkan bahwa alat ini sudah bekerja dengan baik dan sesuai dengan yang diharapkan.

4.3 Pengujian Fungsional Alat

Pengujian fungsional alat yaitu untuk menguji bagian dari alat pemberi pakan ikan ini dapat bekerja atau tidak, maka dibuatlah sebuah tabel 4.3 pengujian :

Tabel 4.3 Hasil Pengujian Fungsional Alat

No.	Daftar Uji	Keterangan	
		Bisa	Tidak Bisa
1.	Tampilan Oled D1 Mini	V	
2.	RTC DS1307	V	
3.	Sensor Infrared	V	
4.	Motor Servo	V	

Pada tabel uji fungsional alat terdapat beberapa bagian yang diuji. Dari hasil tersebut didapatkan bahwa semua bagian pada alat dapat bekerja dengan baik. Bagian-bagian tersebut tidak lepas dari program yang di-downloadkan pada IC mikrokontroler *wemos d1 mini*. Adapun listing program dari bagian-bagian yang diuji adalah sebagai berikut :

4.3.1 Tampilan Oled D1 Mini

Pada program penampilan pada oled d1 mini dimulai dengan tampilan “ Ayu 150120076 “, kemudian dilanjutkan dengan pembacaan RTC DS1307 dengan menampilkan waktu real saat itu juga, dan dilanjutkan dengan tampilan Alarm 1, 2 dan 3, yang terakhir adalah tampilan indikator saat servo open dan close. Adapun listing programnya yaitu :

1. Tampilan Awal

Program ini merupakan program untuk tampilan awal pada oled d1 mini saat alat aktif. Berikut programnya :

PakanInfraredServo1 \$

```
void setup() {  
  #ifndef ESP8266  
    while (!Serial);  
  #endif  
  Wire.begin();  
  Serial.begin(9600);  
  jatayu.begin();  
  rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));  
  if (! rtc.begin()) {  
    Serial.println("RTC tidak ditemukan");  
    while (1);  
  }  
  if (! rtc.isrunning()) {  
    Serial.println("RTC tidak jalan!");  
    rtc.adjust(DateTime(F( __DATE__ ), F( __TIME__ )));  
  }  
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);  
  display.clearDisplay();  
  display.setTextSize(1);  
  display.setTextColor(WHITE);  
  display.setCursor(55, 10);  
  display.println("Ayu");  
  display.display();  
  display.setCursor(40, 20);  
  display.println("150120076");  
  display.display();  
  delay(2000);  
  display.clearDisplay();  
  Servo.attach(servoPin);  
}
```

```
}  
display.begin(SSD1306_SWITCHCAPVCC, 0x3C);  
display.clearDisplay();  
display.setTextSize(1);  
display.setTextColor(WHITE);  
display.setCursor(55, 10);  
display.println("Ayu");  
display.display();  
display.setCursor(40, 20);  
display.println("150120076");
```

```

display.display();
delay(2000);
display.clearDisplay();
Servo1.attach(servoPin);
}

```

2. Tampilan Waktu

Program ini merupakan program untuk menampilkan waktu pada oled d1 mini saat alat aktif. Berikut programnya :

```

void tampil_rtc(){
    sensorValue = analogRead(analogPin); // Read the analog in value:
    DateTime now = rtc.now();
    // Serial.print(now.day(), DEC);
    // Serial.print('/');
    // Serial.print(now.month(), DEC);
    // Serial.print('/');
    // Serial.print(now.year(), DEC);
    // Serial.print(" (");
    // Serial.print(daysOfTheWeek[now.dayOfTheWeek()]);
    // Serial.print(") ");
    // Serial.print(now.hour(), DEC);
    // Serial.print(':');
    // Serial.print(now.minute(), DEC);
    // Serial.print(':');
    // Serial.print(now.second(), DEC);
    // Serial.println();
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.clearDisplay();
    display.setCursor(40, 10);
    display.print(now.day(), DEC);

```

```

display.print('/');
display.print(now.month(), DEC);
display.print('/');
display.print(now.year(), DEC);
display.display();
display.setCursor(40, 20);
display.print(now.hour(), DEC);
display.print(':');
display.print(now.minute(), DEC);
display.print(':');
display.print(now.second(), DEC);
display.display();
strTanggal                                     =
String(now.year()+"-"+String(now.month())+"-"+String(now.day()));
strWaktu                                       =
String(now.hour()+":"+String(now.minute())+"-"+String(now.second()
));
String dataStr = strTanggal+" "+strWaktu;
jatayu.SetData("dataTimer", dataStr);
Serial.println(sensorValue);
}

```

3. Tampilan Alarm

Pada program ini merupakan program untuk menampilkan tampilan alarm pada oled d1 mini saat alat aktif. Berikut programnya :

```

void loop() {
    DateTime now = rtc.now();
    JamAlarm1 = dataV1.substring(0,2);
    MenAlarm1 = dataV1.substring(2,4);
    JamAlarm2 = dataV2.substring(0,2);
    MenAlarm2 = dataV2.substring(2,4);

```

```
JamAlarm3 = dataV3.substring(0,2);
MenAlarm3 = dataV3.substring(2,4);
display.setTextSize(1);
display.setTextColor(WHITE);
display.clearDisplay();
display.setCursor(40, 10);
display.println(" Alarm 1");
display.display();
display.setCursor(40, 20);
display.print(JamAlarm1);
display.print(':');
display.print(MenAlarm1);
display.display();
delay(1000);
```

```
display.setTextSize(1);
display.setTextColor(WHITE);
display.clearDisplay();
display.setCursor(40, 10);
display.println(" Alarm 2");
display.display();
display.setCursor(40, 20);
display.print(JamAlarm2);
display.print(':');
display.print(MenAlarm2);
display.display();
delay(1000);
```

```
display.setTextSize(1);
    display.setTextColor(WHITE);
    display.clearDisplay();
```

```
display.setCursor(40, 10);
display.println(" Alarm 3");
display.display();
display.setCursor(40, 20);
display.print(JamAlarm3);
display.print(':');
display.print(MenAlarm3);
display.display();
delay(1000);
```

```
AlarmJam1 = JamAlarm1.toInt();
AlarmMen1 = MenAlarm1.toInt();
AlarmJam2 = JamAlarm2.toInt();
AlarmMen2 = MenAlarm2.toInt();
AlarmJam3 = JamAlarm3.toInt();
AlarmMen3 = MenAlarm3.toInt();
```

```
String dataV0 = jatayu.GetData("V0");
jam = dataV0.substring(0,2);
men = dataV0.substring(2,4);
det = dataV0.substring(4,6);
d0 = jam.toInt();
d1 = men.toInt();
d2 = det.toInt();
if (dataV0 == "0"){
    jam2 = d0;
    men2 = d1;
}
```

4. Tampilan Servo Open dan Close

Pada program ini merupakan program untuk menampilkan tampilan servo open dan close pada oled d1 mini saat alat aktif. Berikut programnya :

```
void scan_alarm(){
  DateTime now = rtc.now();
  if ((now.hour() == AlarmJam1)&&(now.minute() == AlarmMen1)){
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.clearDisplay();
    display.setCursor(40, 10);
    display.println("Servo");
    display.display();
    display.setCursor(40, 20);
    display.println("Open");
    display.display();
    Servo1.write(0);
    delay(1000);
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.clearDisplay();
    display.setCursor(40, 10);
    display.println("Servo");
    display.display();
    display.setCursor(40, 20);
    display.println("Close");
    display.display();
    Servo1.write(90);
    delay(1000);
  }
}
```

4.3.2 RTC DS1307

RTC DS 1307 berfungsi sebagai pewaktu yang memberikan waktu real, sehingga RTC DS 1307 dapat dikatakan bekerja dengan baik apabila waktu yang keluar benar-benar waktu real saat itu adapun program untuk RTC DS1307 sebagai berikut :

```
#include "RTCLib.h"

//=====
=====

#if defined(ARDUINO_ARCH_SAMD)
    #define Serial SerialUSB
#endif

#include <JatayuIO.h>

String apikey = "f167965e-75f5-11e9-a008-4ec6f6809b89";

char* ssid = "AyAlisrah";
char* password = "desember";

JatayuIO jatayu(apikey, ssid, password);

//=====
=====

RTC_DS1307 rtc;
const int DS1307 = 0x68; // Address of DS1307 see data sheets

char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday"};

const char* days[] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",
"Friday", "Saturday"};
```

```
const char* months[] = {"January", "February", "March", "April", "May", "June",
"July", "August", "September", "October", "November", "December"};

// Initializes all values:
byte second = 0;
byte minute = 0;
byte hour = 0;
byte weekday = 0;
byte monthday = 0;
byte month = 0;
byte year = 0;
String incomingByte[3];
int i = 0;
String jam;
String men;
String det;
int d0;
int d1;
int d2;
int jam2;
int men2;
int h;
int m;
int s;
String strTanggal;
String strWaktu;
String JamAlarm1;
String MenAlarm1;
String JamAlarm2;
String MenAlarm2;
String JamAlarm3;
```

```

String MenAlarm3;
int AlarmJam1;
int AlarmMen1;
int AlarmJam2;
int AlarmMen2;
int AlarmJam3;
int AlarmMen3;

void setup() {
  #ifndef ESP8266
    while (!Serial);
  #endif
  Wire.begin();
  Serial.begin(9600);
  jatayu.begin();
  rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
  if (! rtc.begin()) {
    Serial.println("RTC tidak ditemukan");
    while (1);
  }
  if (! rtc.isrunning()) {
    Serial.println("RTC tidak jalan!");
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
  }

void loop() {
  DateTime now = rtc.now();
  String dataV1 = jatayu.GetData("V1");
  String dataV2 = jatayu.GetData("V2");
  String dataV3 = jatayu.GetData("V3");
  JamAlarm1 = dataV1.substring(0,2);
  MenAlarm1 = dataV1.substring(2,4);

```

```
JamAlarm2 = dataV2.substring(0,2);
MenAlarm2 = dataV2.substring(2,4);
JamAlarm3 = dataV3.substring(0,2);
MenAlarm3 = dataV3.substring(2,4);
display.setTextSize(1);
display.setTextColor(WHITE);
display.clearDisplay();
display.setCursor(40, 10);
display.println("Alarm 1");
display.display();
display.setCursor(40, 20);
display.print(JamAlarm1);
display.print(':');
display.print(MenAlarm1);
display.display();
delay(1000);
```

```
display.setTextSize(1);
display.setTextColor(WHITE);
display.clearDisplay();
display.setCursor(40, 10);
display.println("Alarm 2");
display.display();
display.setCursor(40, 20);
display.print(JamAlarm2);
display.print(':');
display.print(MenAlarm2);
display.display();
delay(1000);
```

```
display.setTextSize(1);
```

```

display.setTextColor(WHITE);
display.clearDisplay();
display.setCursor(40, 10);
display.println("Alarm 3");
display.display();
display.setCursor(40, 20);
display.print(JamAlarm3);
display.print(':');
display.print(MenAlarm3);
display.display();
delay(1000);
//Serial.print("jatayu.GetData");
//Serial.println(dataV1);
//Serial.print("jatayu.GetData");
//Serial.println(dataV2);
//Serial.print("jatayu.GetData");
//Serial.println(dataV3);

AlarmJam1 = JamAlarm1.toInt();
AlarmMen1 = MenAlarm1.toInt();
AlarmJam2 = JamAlarm2.toInt();
AlarmMen2 = MenAlarm2.toInt();
AlarmJam3 = JamAlarm3.toInt();
AlarmMen3 = MenAlarm3.toInt();

String dataV0 = jatayu.GetData("V0");
jam = dataV0.substring(0,2);
men = dataV0.substring(2,4);
det = dataV0.substring(4,6);
d0 = jam.toInt();
d1 = men.toInt();

```

```

d2 = det.toInt();
if (dataV0 == "0"){
    jam2 = d0;
    men2 = d1;
}
else if (dataV0 == ""){
    Serial.println("data error");
    delay(5000);
}
else{
    jam2 = d0 - now.hour();
    men2 = d1 - now.minute();
    jatayu.SetData("V0", 0);
    setTime();

    String dataV1 = jatayu.GetData("V1");
    String dataV2 = jatayu.GetData("V2");
    String dataV3 = jatayu.GetData("V3");
    JamAlarm1 = dataV1.substring(0,2);
    MenAlarm1 = dataV1.substring(2,4);
    JamAlarm2 = dataV2.substring(0,2);
    MenAlarm2 = dataV2.substring(2,4);
    JamAlarm3 = dataV3.substring(0,2);
    MenAlarm3 = dataV3.substring(2,4);
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.clearDisplay();
    display.setCursor(40, 10);
    display.println("Alarm 1");
    display.display();
    display.setCursor(40, 20);
    display.print(JamAlarm1);

```

```
display.print(':');  
display.print(MenAlarm1);  
display.display();  
delay(1000);
```

```
display.setTextSize(1);  
display.setTextColor(WHITE);  
display.clearDisplay();  
display.setCursor(40, 10);  
display.println("Alarm 2");  
display.display();  
display.setCursor(40, 20);  
display.print(JamAlarm2);  
display.print(':');  
display.print(MenAlarm2);  
display.display();  
delay(1000);
```

```
display.setTextSize(1);  
display.setTextColor(WHITE);  
display.clearDisplay();  
display.setCursor(40, 10);  
display.println("Alarm 3");  
display.display();  
display.setCursor(40, 20);  
display.print(JamAlarm3);  
display.print(':');  
display.print(MenAlarm3);  
display.display();  
delay(1000);
```



```

    AlarmJam1 = JamAlarm1.toInt();
    AlarmMen1 = MenAlarm1.toInt();
    AlarmJam2 = JamAlarm2.toInt();
    AlarmMen2 = MenAlarm2.toInt();
    AlarmJam3 = JamAlarm3.toInt();
    AlarmMen3 = MenAlarm3.toInt();
}
tampil_rtc();
scan_alarm();
}

void tampil_rtc(){
    sensorValue = analogRead(analogPin); // Read the analog in value:
    DateTime now = rtc.now();
    // Serial.print(now.day(), DEC);
    // Serial.print('/');
    // Serial.print(now.month(), DEC);
    // Serial.print('/');
    // Serial.print(now.year(), DEC);
    // Serial.print(" ");
    // Serial.print(daysOfTheWeek[now.dayOfTheWeek()]);
    // Serial.print(" ");
    // Serial.print(now.hour(), DEC);
    // Serial.print(':');
    // Serial.print(now.minute(), DEC);
    // Serial.print(':');
    // Serial.print(now.second(), DEC);
    // Serial.println();
    display.setTextSize(1);
    display.setTextColor(WHITE);

```

```

display.clearDisplay();
display.setCursor(40, 10);
display.print(now.day(), DEC);
display.print('/');
display.print(now.month(), DEC);
display.print('/');
display.print(now.year(), DEC);
display.display();
display.setCursor(40, 20);
display.print(now.hour(), DEC);
display.print(':');
display.print(now.minute(), DEC);
display.print(':');
display.print(now.second(), DEC);
display.display();
strTanggal =
String(now.year()+"/"+String(now.month()+"/"+String(now.day()));
strWaktu =
String(now.hour()+":"+String(now.minute()+":"+String(now.second()));
String dataStr = strTanggal+" "+strWaktu;
jatayu.SetData("dataTimer", dataStr);
jatayu.SetData("dataInfrared", sensorValue);
Serial.print("dataInfrared");
Serial.println(sensorValue);

}

```

4.3.3 Sensor Infrared

Sensor infrared berfungsi untuk mengetahui ada dan tidaknya pakan ikan didalam wadah penampungan, agar dapat diketahui telah dibuat sebuah aplikasi

khusus untuk memonitoring pakan ikan dalam wadah. Adapun programnya sebagai berikut :

```
const int analogPin = A0; // Analog input pin 0 (GPIO 36)
int sensorValue = 0; // Value read from the ADC
```

Dari program diatas dapat diketahui bahwa pin yang digunakan yaitu pin A0

```
    jatayu.SetData("dataTimer", dataStr);
    jatayu.SetData("dataInfrared", sensorValue);
    Serial.print("dataInfrared");
    Serial.println(sensorValue);
```

Program diatas merupakan program yang bertujuan agar hasil dari sensor infrared dapat terbaca pada aplikasi dan dapat dimonitor lewat serial monitor pada Arduino IDE.

4.3.4 Motor Servo

Motor servo akan dikatakan bekerja dengan baik apabila saat tiba waktu alarm yang ditentukan maka motor servo akan membuka tutup tempat keluarnya pakan ikan dalam wadah penampungan. Dan saat servo bekerja akan ditampilkan pada oled d1 mini. Adapun programnya adalah :

1. Memanggil library :

```
#include <Servo.h>
int servoPin = D8;
Servo Servo1;
```

2. Program servo open dan close :

```
void scan_alarm(){
```

```

DateTime now = rtc.now();
if ((now.hour() == AlarmJam1)&&(now.minute() ==
AlarmMen1)){
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.clearDisplay();
    display.setCursor(40, 10);
    display.println("Servo");
    display.display();
    display.setCursor(40, 20);
    display.println("Open");
    display.display();
    Servo1.write(180);
    delay(1000);
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.clearDisplay();
    display.setCursor(40, 10);
    display.println("Servo");
    display.display();
    display.setCursor(40, 20);
    display.println("Close");
    display.display();
    Servo1.write(90);
    delay(1000);
}
if ((now.hour() == AlarmJam2)&&(now.minute() ==
AlarmMen2)){
display.setTextSize(1);
display.setTextColor(WHITE);
display.clearDisplay();

```

```

display.setCursor(40, 10);
display.println("Servo");
display.display();
display.setCursor(40, 20);
display.println("Open");
display.display();
Servo1.write(180);
delay(1000);
display.setTextSize(1);
display.setTextColor(WHITE);
display.clearDisplay();
display.setCursor(40, 10);
display.println("Servo");
display.display();
display.setCursor(40, 20);
display.println("Close");
display.display();
Servo1.write(90);
delay(1000);
}
if ((now.hour() == AlarmJam3)&&(now.minute() == AlarmMen3)){
display.setTextSize(1);
display.setTextColor(WHITE);
display.clearDisplay();
display.setCursor(40, 10);
display.println("Servo");
display.display();
display.setCursor(40, 20);
display.println("Open");
display.display();
Servo1.write(180);

```

```

        delay(1000);
        display.setTextSize(1);
        display.setTextColor(WHITE);
        display.clearDisplay();
        display.setCursor(40, 10);
        display.println("Servo");
        display.display();
        display.setCursor(40, 20);
        display.println("Close");
        display.display();
        Servo1.write(90);
        delay(1000);
    }
}
byte decToBcd(byte val) {
    return ((val/10*16) + (val% 10));
}
byte bcdToDec(byte val) {
    return ((val/16*10) + (val% 16));
}

```

4.4 Pengujian Jadwal Otomatisasi Alat Pemberi Pakan

Pengujian ini dilakukan agar dapat diketahui apakah alat setiap harinya bekerja dengan baik dan benar sesuai dengan jadwal yang sudah ditentukan/diatur. Didalam pengujian ini diambil data selama 4 hari dan setiap harinya terdapat 3 jadwal pemberian pakan ikan yaitu waktu pagi, siang dan malam. Berikut adalah tabel pengujiannya :

Tabel 4.4 Hasil Pengujian Jadwal otomatis pada alat di waktu pagi

No.	Percobaan Hari Ke-	Jadwal waktu yang diatur pada aplikasi	Waktu Motor Servo Bekerja	Delay antara waktu pada aplikasi dan motor servo	Status
1.	Hari Pertama	08.30.00	08.29.58	2 detik	Bekerja
2.	Hari Kedua	08.30.00	08.29.59	1 detik	Bekerja
3.	Hari Ketiga	08.30.00	08.29.59	1 detik	Bekerja
4.	Hari Keempat	08.30.00	08.29.58	2 detik	Bekerja

Pada tabel 4.4 diatas dari hasil pengujian jadwal otomatisasi, selama 4 hari selalu bekerja sesuai dengan penjadwalan pemberian pakan pada pagi hari. Pada hari pertama waktu jadwal dan waktu alat sebesar 2 detik. Hal ini dikarenakan server aplikasi mengirim perintah selalu lebih cepat sebelum jadwal sehingga alat bekerja 1 – 3 detik lebih cepat dibandingkan dengan penjadwalan pemberian pakan yang diatur pada aplikasi.

Tabel 4.5 Hasil Pengujian Jadwal Otomatis pada alat di waktu siang

No.	Percobaan Hari Ke-	Jadwal waktu yang diatur pada aplikasi	Waktu Motor Servo Bekerja	Delay antara waktu pada aplikasi dan motor servo	Status
1.	Hari Pertama	13.30.00	13.29.58	2 detik	Bekerja
2.	Hari Kedua	13.30.00	13.29.57	3 detik	Bekerja
3.	Hari Ketiga	13.30.00	13.29.59	1 detik	Bekerja

No.	Percobaan Hari Ke-	Jadwal waktu yang diatur pada aplikasi	Waktu Motor Servo Bekerja	Delay antara waktu pada aplikasi dan motor servo	Status
4.	Hari Keempat	13.30.00	13.29.58	2 detik	Bekerja

Pada tabel 4.5 diatas dari hasil pengujian jadwal otomatis, selama 4 hari alat selalu bekerja sesuai dengan penjadwalan pemberian pakan pada siang hari. Pada hari pertama waktu jadwal sebesar 2 detik. Hal ini dikarenakan server aplikasi mengirim perintah selalu lebih cepat sebelum jadwal sehingga alat bekerja 1 – 3 detik lebih cepat dibandingkan dengan penjadwalan pemberian pakan yang diatur pada aplikasi.

Tabel 4.6 Hasil Pengujian Jadwal Otomatis pada alat di waktu malam

No.	Percobaan Hari Ke-	Jadwal waktu yang diatur pada aplikasi	Waktu Motor Servo Bekerja	Delay antara waktu pada aplikasi dan motor servo	Status
1.	Hari Pertama	22.30.00	22.29.59	1 detik	Bekerja
2.	Hari Kedua	22.30.00	22.29.59	1 detik	Bekerja
3.	Hari Ketiga	22.30.00	22.29.59	1 detik	Bekerja
4.	Hari Keempat	22.30.00	22.29.58	2 detik	Bekerja

Pada tabel 4.6 diatas dari hasil pengujian jadwal otomatis, selama 4 hari alat selalu bekerja sesuai dengan penjadwalan pemberian pakan pada malam hari. Pada hari pertama waktu jadwal sebesar 1 detik. Hal ini dikarenakan server aplikasi mengirim perintah selalu lebih cepat sebelum jadwal sehingga alat bekerja 1 – 3

detik lebih cepat dibandingkan dengan penjadwalan pemberian pakan yang diatur pada aplikasi

4.5 Pengujian Berat Pakan yang dikeluarkan

Pengamatan dilakukan dengan mengukur berat pakan menggunakan timbangan digital, selanjutnya dengan menghitung berat rata-rata dan didapatkan selisih atau presentase kesalahannya. Hasil rata-rata berat pakan yang diamati dibedakan menjadi 3 bagian, yaitu untuk waktu buka tutup motor servo 500ms, 1000ms, dan 1500ms.

4.5.1 Pengujian berat pakan yang dikeluarkan dengan waktu buka tutup motor servo 500ms

Tabel 4.7 Pengujian berat pakan yang dikeluarkan dengan waktu buka tutup motor servo 500ms

Percobaan	Berat Pakan (gram)
1	4
2	3
3	4
4	4
5	3

Hasil rata-rata berat pakan yang dikeluarkan dengan waktu buka tutup motor servo 500ms yaitu :

$$\frac{4+3+4+4+3}{5} = 3,6 \text{ gr} \quad (3.1)$$

Selanjutnya yaitu perhitungan persentase kesalahan dalam lima kali percobaan dengan waktu buka tutup motor servo 500ms.

$$\text{Dengan berat pakan 4} = \frac{|4-3,6|}{3,6} \times 100\% = 11,11\% \quad (3.3)$$

$$\text{Dengan berat pakan 3} = \frac{|3-3,6|}{3,6} \times 100\% = 16,67\% \quad (3.3)$$

$$\text{Dengan berat pakan 4} = \frac{|4-3,6|}{3,6} \times 100\% = 11,11\% \quad (3.3)$$

$$\text{Dengan berat pakan 4} = \frac{|4-3,6|}{3,6} \times 100\% = 11,11\% \quad (3.3)$$

$$\text{Dengan berat pakan 3} = \frac{|3-3,6|}{3,6} \times 100\% = 16,67\% \quad (3.3)$$

Persentase kesalahan diatas yaitu 11,11%, 16,67%, 11,11%, 11,11%, dan 16,67%.

Dari hasil perhitungan tersebut didapatkan hasil rata-rata persentase kesalahan

$$\text{Dengan waktu buka 500ms} = \frac{11,11+16,67+11,11+11,11+16,67}{5} = 13,33\% \quad (3.1)$$

4.5.2 Pengujian berat pakan yang dikeluarkan dengan waktu buka tutup motor servo 1000ms

Tabel 4.8 Pengujian berat pakan yang dikeluarkan dengan waktu buka tutup motor servo 1000ms

Percobaan	Berat Pakan (gram)
1	8
2	8
3	7
4	8
5	8

Hasil rata-rata berat pakan yang dikeluarkan dengan waktu buka tutup motor servo 1000ms yaitu :

$$\frac{8+8+7+8+8}{5} = 7,8 \text{ gr} \quad (3.1)$$

Selanjutnya yaitu perhitungan persentase kesalahan dalam lima kali percobaan dengan waktu buka tutup motor servo 1000ms.

$$\text{Dengan berat pakan 8} = \frac{|8-7,8|}{7,8} \times 100\% = 2,56\% \quad (3.3)$$

$$\text{Dengan berat pakan 8} = \frac{|8-7,8|}{7,8} \times 100\% = 2,56\% \quad (3.3)$$

$$\text{Dengan berat pakan 7} = \frac{|7-7,8|}{7,8} \times 100\% = 10,26\% \quad (3.3)$$

$$\text{Dengan berat pakan 8} = \frac{|8-7,8|}{7,8} \times 100\% = 2,56\% \quad (3.3)$$

$$\text{Dengan berat pakan 8} = \frac{|8-7,8|}{7,8} \times 100\% = 2,56\% \quad (3.3)$$

Persentase kesalahan diatas yaitu 2,56%, 2,56%, 10,26%, 2,56% dan 2,56%. Dari hasil perhitungan tersebut didapatkan hasil rata-rata persentase kesalahan

$$\text{Dengan waktu buka 1000ms} = \frac{2,56+2,56+10,26+2,56+2,56}{5} = 4,1\% \quad (3.1)$$

4.5.3 Pengujian berat pakan yang dikeluarkan dengan waktu buka tutup motor servo 1500ms

Tabel 4.9 Pengujian berat pakan yang dikeluarkan dengan waktu buka tutup motor servo 1500ms

Percobaan	Berat Pakan (gram)
1	14
2	13
3	13
4	14
5	14

Hasil rata-rata berat pakan yang dikeluarkan dengan waktu buka tutup motor servo 1500ms yaitu :

$$\frac{14+13+13+14+14}{5} = 13,6 \text{ gr} \quad (3.1)$$

Selanjutnya yaitu perhitungan persentase kesalahan dalam lima kali percobaan dengan waktu buka tutup motor servo 1500ms.

$$\text{Dengan berat pakan 14} = \frac{|14-13,6|}{13,6} \times 100\% = 2,94\%$$

(3.3)

$$\text{Dengan berat pakan 13} = \frac{|13-13,6|}{13,6} \times 100\% = 4,41\%$$

(3.3)

$$\text{Dengan berat pakan 13} = \frac{|13-13,6|}{13,6} \times 100\% = 4,41\%$$

(3.3)

$$\text{Dengan berat pakan 14} = \frac{|14-13,6|}{13,6} \times 100\% = 2,94\%$$

(3.3)

$$\text{Dengan berat pakan 14} = \frac{|14-13,6|}{13,6} \times 100\% = 2,94\%$$

(3.3)

Persentase kesalahan diatas yaitu 2,94%, 4,41%, 4,41%, 2,94%, dan 2,94%. Dari hasil perhitungan tersebut didapatkan hasil rata-rata persentase kesalahan

$$\text{Dengan waktu buka 1500ms} = \frac{2,94+4,41+4,41+2,94+2,94}{5} = 3,53\% \quad (3.1)$$

Hasil perhitungan diatas dapat membedakan berat pakan dalam 3 jenis waktu buka motor servo. Pertama pada waktu buka motor servo 500ms, berat pakan yang dikeluarkan rata-rata 3,6 gram dengan presentase kesalahan rata-rata sebesar 13,33%. Kedua pada waktu buka motor servo 1000ms, berat pakan yang dikeluarkan rata-rata yaitu sebanyak 7,8 gram dengan presentase kesalahan 4,1%. Dan ketiga pada waktu buka motor servo 1500ms, berat pakan yang dikeluarkan rata-rata yaitu sebesar 13,6 gram, dengan presentase kesalahan 3,53%. Dari hasil tersebut dapat disimpulkan bahwa saat waktu buka motor servo cepat yaitu 500ms maka pakan ikan yang keluarpun sedikit dan sebaliknya begitu jika waktu buka servo lama yaitu 1500ms maka pakan ikan yang keluarpun banyak jumlahnya. Ini dipengaruhi oleh waktu buka motor servonya.

4.6 Monitoring Level Pakan

Proses monitoring level pakan melalui aplikasi *SetAlarm* di Android dengan menampilkan ketinggian pakan ikan pada wadah penampungan. Didalam aplikasi *SetAlarm* terdapat indicator yaitu feed level dimana itu menunjukkan ketinggian pakan ikan dalam wadah penampungan. Saat alat aktif maka secara otomatis sensor infrarednya akan bekerja dan membaca jarak maka dengan itu akan diketahui berapa nilai feed levelnya setiap centimeternya. Berikut adalah tabel perhitungannya :

Tabel 4.10 Hasil Pembacaan Nilai Feed Level Percentimeter

No.	Nilai CM (Jarak)	Nilai Feed Level (Range)
1.	5cm	732 - 744
2.	6cm	629 - 636
3.	7cm	548 - 555
4.	8cm	488 – 498
5.	9cm	435 - 440
6.	10cm	389 - 393
7.	11cm	354 – 365
8.	12cm	325 - 334
9.	13cm	301 – 308
10.	14cm	282 – 290
11.	15cm	264 - 272

Pada alat otomatisasi pemberian pakan ikan memiliki ketinggian 15cm, dimana pada saat pakan dimasukan kedalam wadah penampungan dan pakan dalam kondisi penuh dalam wadah maka akan berada pada ketinggian 5cm dan akan

terbaca pada aplikasi nilai feed levelnya berada diantara nilai 732 – 744, sebaliknya saat kondisi pakan ikan habis maka akan terbaca dijarak/ketinggian 15cm dan akan terbaca pada aplikasi *SetAlarm* nilai feed levelnya berada diantara 264 – 272. Dan bisa dilihat pada gambar 4.1 :



Gambar 4.1 Tampilan Feed Level dalam Aplikasi

4.7 Kinerja Alat

Setelah melakukan proses pembuatan, *literature* serta perencanaan, pengukuran alat dan perhitungan maka, penulis dapat menyimpulkan sebagai berikut:

1. Berdasarkan data yang diperoleh saat pengujian catu daya dapat berfungsi dengan baik dapat dilihat dari tabel 4.1 dan berdasarkan hasil perhitungan *error*nya sebesar 1% saat kondisi *standby* dan 3% saat kondisi motor servo

bekerja. Sedangkan pada pengujian fungsional alat semua dapat bekerja dengan baik dapat dilihat dari tabel 4.3.

2. Berdasarkan pengujian jadwal otomatisasi alat pemberi pakan ikan berfungsi dengan baik dimana alat setiap harinya bekerja dengan baik dan benar sesuai dengan jadwal yang sudah ditentukan/diatur, dapat dilihat dari tabel 4.4, 4.5 dan 4.6.
3. Berdasarkan pengujian berat pakan yang dikeluarkan dibagi menjadi 3 bagian yaitu saat buka tutup motor servo 500ms dimana didapat hasil rata-rata *errornya* 13,33%, saat buka tutup motor servo 1000ms memiliki hasil rata-rata *errornya* 2,56%, dan saat buka tutup motor servo 1500ms memiliki hasil rata-rata *errornya* 2,94 %. Tetapi jumlah pakan yang keluar paling banyak saat buka tutup motor servo 1500ms dapat dilihat dari tabel 4.9 dimana hasil rata-ratanya sebesar 13,6gr dan yang terkecil jumlah pakannya adalah saat buka tutup motor servo 500ms dapat dilihat dari tabel 4.7 dimana hasil rata-ratanya sebesar 500ms. Jumlah pakan ikan yang keluar dapat mempengaruhi kekeruhan air dan menyebabkan ikan yang ada dalam akuarium dapat mati setelah dilakukan sebuah pengujian langsung diakuarium saat buka tutup servo 1000ms dengan rata-rata hasil berat pakan yang keluar sebesar 7,8gr membuat akuarium menjadi keruh karena cukup banyaknya pakan yang keluar dan jumlah ikan didalam akuarium yang hanya 2 ekor sehingga membuat salah satu ikan mati.
4. Berdasarkan monitoring level pakan berfungsi dengan baik dimana pakan ikan dapat dideteksi apakah telah habis ataupun masih cukup banyak didalam wadah, dapat dilihat dari tabel 4.10 dimana tabel tersebut merupakan nilai feed level per centimeter, feed level sendiri dimonitoring melalui aplikasi dan dapat dilihat pada gambar 4.1.

4.8 Keunggulan dan Kelemahan Alat

Berikut adalah Keunggulan alat otomatisasi pemberi pakan ikan berbasis *internet of things* (IOT) :

1. Kontrol menggunakan aplikasi yang dibuat sendiri menggunakan MIT App Inventor sehingga bisa dioperasikan lewat jarak jauh dan lebih efektif dalam proses penggunaannya
2. Alat ini dapat mengatur waktu pemberian pakan ikan sehingga saat waktu ikan untuk makan maka alat akan bekerja memberikan makan ikan.
3. Terdapat display LCD, untuk menampilkan waktu dan alarm yang telah *disetting*.

Berikut adalah kelemahan alat otomatisasi pemberi pakan ikan berbasis *internet of things* (IOT) :

1. Alat belum dilengkapi dengan *backup* catu daya cadangan sehingga jika terjadi gangguan dari PLN seperti pemadaman listrik maka alat ini tidak akan bekerja sebagaimana mestinya.
2. Alat ini hanya dapat dipasang pada akuarium ukuran tertentu, sehingga diperlukan pengembangan mekanik agar alat dapat digunakan dengan berbagai jenis ukuran akuarium.
3. Berat pakan yang dikeluarkan dengan waktu buka motor servo yang berbeda masih belum linear berdasarkan hasil pengujian. Dan jumlah pakan yang keluar mempengaruhi tingkat kekeruhan air didalam akuarium sehingga dapat mengakibatkan ikan didalam akuarium dapat mati apabila air didalam akuarium tersebut keruh.
4. Alat ini dapat bekerja dan dikontrol apabila tersambung diinternet, jika koneksi internet tidak stabil maka pembacaan alat pada aplikasi tersebut akan *error*, untuk mengontrol alat melalui aplikasi pada android akan mengalami gangguan dimana akan muncul notifikasi pada android bahwa sistem *error*, alampun tidak bisa dikontrol penjadwalannya.