DAFTAR PUSTAKA

Admin. (2020). Analisa Kualitas Air Budidaya Ikan: Parameter Kimia, (Online), (http://www.saka.co.id/news detail/analisa kualitas-air-budidayaikan--parameter-kimia).

Andriadhi, A., Bambang, A. N., & Darmanto, Y. S. (2016). STRATEGI PENGEMBANGAN BUDIDAYA IKAN HIAS AIR TAWAR DI KOTA SEMARANG *(The Development Strategy Of Freshwater Ornamental Fish Culture In Semarang)*.

Andriani, Y, Maesaroh, T.R.S., Yustiati, A, Iskandar, dan Zidni, I. (2018). Kualitas warna benih ikan mas koki (Carassius auratus) Oranda pada berbagai tingkat pemberian tepung Spirulina plantesis. Chimica et Natura Acta. 6(2), 49-55.

Arduino. (2024). Guide to DS18B20 Temperature Sensor with Arduino. Arduino.cc. Diakses dari https://www.arduino.cc

Aulia M. T. dan Nani Anisah. (2022). ”Sistem Kontrol dan Monitoring Kualitas Air Pada Budidaya Ikan Lele dengan Media Kolam Berbasis IoT”. Proyek Akhir tidak diterbitkan. Bangka Belitung: Diploma III Politeknik Manufaktur Negeri Bangka Belitung.

Autodesk Fusion 360 Overview. Autodesk. Diakses dari https://www.autodesk.com /products/fusion-360/overview.

Bahari, W., & Sugiarto, A. (2019). Rancang bangun alat pendeteksi kebakaran berbasis Internet of Things (IoT). *Doctoral Dissertation*.

Boulanger, D. (2016). Arduino Workshop: A Hands-On Introduction with 65 Projects. San Francisco: No Starch Press.

Dinas Kelautan Dan Perikanan Kabupaten Tulungagung. (2014). Data Dan Informasi Perikanan Kabupaten Tulungagung Tahun 2014. Tulungagung : Dinas Kelautan Dan Perikanan Kabupaten Tulungagung

DFRobot. (2023, April 24). *Gravity Analog TDS Sensor Meter For Arduino SKU*. Retrieved July 2024, from https://wiki.dfrobot.com/Gravity__Analog_ TDS_Sensor___Meter_For_Arduino_SKU__SEN0244

Fazil, M., Adhar, S., & Ezraneti, R. (2017). Efektivitas penggunaan ijuk, jerami padi dan ampas tebu sebagai filter air pada pemeliharaan ikan mas koki (Carassius Auratus). Acta Aquatica, 4, 1(April) 37-43.

Imam, T. (2010). Uji Multi Lokasi Pada Budidaya Ikan Nila dengan Sistem Akuaponik. Laporan Hasil Penelitian. Badan Riset Kelautan dan Perikanan: Jakarta, 30.

Instructables. (2023). DIY Water Level Monitoring Using JSN-SR04T. Diakses dari https://www.instructables.com.

Irwansyah, M. (2019). Sistem Kontrol Air dalam Akuarium Modern. Bandung: Aquatic Press.

Kafuku, T., & Ikenoue, T. (1983). Nutritional ecology of goldfish (Carassius auratus). Aquaculture, 35(1), 1-16.

Lutfiyah. (2022). Monitor biaya listrik berbasis IoT (internet of things).

MathWorks. (2020). *MATLAB Documentation*. Retrieved from https://www.mathworks.com/help/matlab

Maxim Integrated. (2023). DS3231: Extremely Accurate I2C-Integrated RTC. Diakses dari https://www.maximintegrated.com.

Monk, S. (2017). Programming Arduino: Getting Started with Sketches (Second Edition). New York: McGraw-Hill Education.

Muchlisin, Z.A., F. Afrido, T. Murda, N. Fadli, A.A. Muhammadar, Z. Jalil, C.Yulvizar. (2016). *The effectiveness of experimental diet with varying levelsofpapain on the growth performance, survival rate and feed utilization of keurelingfish* (Tor tambra). Biosaintifika, 8: 172-177.

Muhammad Irwansyah, D. (2013). Pompa air aquarium menggunakan solar panel. *Jurnal Integrasi*, 85-90.

Mustafa Arief. (2020). Pengelolaan Kualitas Air untuk Akuakultur. Jepara: UNISNU Press

Nurwachid, Y. (2023). Prototype Sistem Pemantauan Kualitas Air Pada Aquascape Menggunakan Nodemcu Esp8266 Berbasis Internet Of Things (IOT).

Pratama, A. (2018). Pengenalan MATLAB dalam Pemrograman Teknik. Yogyakarta: Penerbit Teknik.

Rahmat A. (2009). Budidaya Ikan Mas Koki. Jakarta: Penebar Swadaya.

RoboDyn. (2024). JSN-SR04T Technical Specification. Diakses dari https://www.robodyn.com.

Salsabila M. dan Hari Suprapto. (2018). ”Teknik Pembesaran Ikan Nila *(Oreochromis Niloticus)* di Instalasi Budidaya Air Tawar Pandaan, Jawa Timur”. *Journal of Aquaculture and Fish Health*, 7 (13), Halaman 121- 122.

Saputra, O., & Ramelan, U. (2018). Analisis efektivitas konversi pompa air model motor penggerak AC dengan pompa air model motor penggerak DC. *Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST)*, 1-7.

Saputra, D., & Ramelan, S. (2019). Panduan Lengkap Pemeliharaan Akuarium. Jakarta: Penerbit Alam.

Sori. (2013). Pengendalian miniatur instalasi Listrik menggunakan mikrokontroller Arduino Mega terintegrasi HMI *(Human Machine Interface)*. Skripsi. Bandung :Jurusan Teknik Elektro. Universitas Pendidikan Indonesia.

SparkFun. (2024). *Real-Time Clock Module Hookup Guide*. Diakses dari https:// www.sparkfun.com.

Susanto, M., Supriyono, E., & Warlina, L. (2017). Strategi budidaya ikan koki baster *(Carassius auratus)* ramah lingkungan dalam upaya peningkatan produksi. Jurnal Matematika Sains dan Teknologi, 12 (2), 133-142.

Syukhron, I., Rahmadewi, R., & Ibrahim. (2021). Penggunaan aplikasi blynk untuk monitoring dan kontrol jarak jauh pada sistem kompos pintar berbasis IoT. *Jurnal Rekayasa dan Teknologi Elektro, 15(1)*, 1-11.

Willem H. Siegers, Yudi Prayitno dan Annita Sari. (2019). Pengaruh Kualitas Air Terhadap Pertumbuhan Ikan Nila Nirwana (Oreochromis Sp.) Pada Tambak Payau. The Journal of Fisheries Development, 3 (2), 95-104.

Zadeh, L. A. (1965). Fuzzy sets. Information and Control, 8(3), 338-353.

Zulfikar, F. (2020). Pemrograman Mikrokontroler Menggunakan Arduino IDE. Jakarta: Penerbit Teknologi.

LAMPIRAN

Lampiran 1 Program Pengujian Sensor Suhu

```
#include <OneWire.h> //library OneWire
#include <DallasTemperature.h> //library DS18B20
#include <LiquidCrystal_I2C.h>

#define ONE_WIRE_BUS 4
OneWire oneWire(ONE_WIRE_BUS);
LiquidCrystal_I2C lcd(0x27, 20, 4);

// mereferensikan sensor ke library one wire
DallasTemperature sensors(&oneWire);
float temperature;
const int heaterRelayPin = 8;
const int fanRelayPin = 11;

void setup(void)
{
  Serial.begin(9600);
  sensors.begin();  // Start sensor
  lcd.begin();
  pinMode(heaterRelayPin, OUTPUT);
  pinMode(fanRelayPin, OUTPUT);
}
void loop(void)
{
 sensors.requestTemperatures();
```

```
 temperature = sensors.getTempCByIndex(0);
 Serial.print("Suhu: ");
 Serial.println(sensors.getTempCByIndex(0));
 lcd.setCursor(0, 1);
 lcd.print("Suhu: ");
 lcd.setCursor(6, 1);
 lcd.print(temperature,1);
 lcd.print((char)223);
 lcd.print("C");
delay(1000);//tunda 1 detik
 // Atur cooler berdasarkan suhu
if (temperature > 30.00) {
 digitalWrite(fanRelayPin, LOW); // Nyalakan kipas saat suhu >
 lcd.setCursor(0, 2);
 lcd.print("Fan: 1");
 }
else if (temperature <= 30.00) {
 digitalWrite(fanRelayPin, HIGH); // Matikan kipas saat suhu < 30

 lcd.setCursor(0, 2);
 lcd.print("Fan: 0");
 }
 // Atur heater berdasarkan suhu
if (temperature < 24.00) {
 digitalWrite(heaterRelayPin, LOW); // Nyalakan heater saat suhu < 24
 lcd.setCursor(0, 3);
 lcd.print("Heater: 1");
 }
else if (temperature >= 24.00) {
```

```
  digitalWrite(heaterRelayPin, HIGH); // Matikan heater saat suhu > 24
  lcd.setCursor(0, 3);
  lcd.print("Heater: 0");
 }
}
```

Lampiran 2 Program Pengujian Sensor TDS

```
#include <EEPROM.h>
#include "GravityTDS.h"
#include <LiquidCrystal_I2C.h>

#define TdsSensorPin A1
GravityTDS gravityTds;
LiquidCrystal_I2C lcd(0x27, 20, 4);

float tdsValue = 0;

void setup()
{
    Serial.begin(9600);
    gravityTds.setPin(TdsSensorPin);
    gravityTds.setAref(5.0);  //reference voltage on ADC, default 5.0V on Arduino
UNO
    gravityTds.setAdcRange(1024);  //1024 for 10bit ADC;4096 for 12bit ADC
    gravityTds.begin();  //initialization
    lcd.begin();
}

void loop()
```

```
{
  gravityTds.update();
  gravityTds.update();
  tdsValue= (gravityTds.getTdsValue()-46.63) ;
  Serial.print(tdsValue,2);
  Serial.println("ppm");
  delay(1000);
  lcd.setCursor(10, 1);
  lcd.print("TDS:");
  lcd.setCursor(14, 1);
  lcd.print(tdsValue);
  lcd.setCursor(17, 1);
  lcd.print("ppm");
}
```

Lampiran 3 Program Pengujian Sensor pH

```
#include <LiquidCrystal_I2C.h>
const int phSensorPin  = A2;
float phValue = 0;

LiquidCrystal_I2C lcd (0x27, 20, 4);

void setup()
{
  Serial.begin(9600);
  lcd.begin() ;
}
```

```
void loop()
{
  int nilaiPengukuranPh = analogRead(phSensorPin);
  Serial.print("Nilai ADC Ph: ");
  Serial.println(nilaiPengukuranPh);
  double TeganganPh = 5 / 1024.0 * nilaiPengukuranPh;
  Serial.print("TeganganPh: ");
  Serial.println(TeganganPh, 3);
  //Po = 7.00 + ((teganganPh7 - TeganganPh) / PhStep);
  //PH7 = 3,03 , PH4 = 3,37
  phValue = 7.00 + ((3.03 - TeganganPh) / 0.12);
  Serial.print("Nilai PH cairan: ");
  Serial.println(phValue, 3);
  delay(1000);
  lcd.setCursor(11, 2);
 lcd.print("PH:");
 lcd.setCursor(14, 2);
 lcd.print(phValue, 2);
}
```

Lampiran 4 Program Kalibrasi Sensor pH

```
void setup()
{
 Serial.begin(9600);
}

void loop()
{
 int sensorValue = analogRead(A2);
 float voltage = sensorValue * (5.0 / 1024.0);
 Serial.println(voltage);
 delay(1000);
}
```

Lampiran 5 Program Pengujian Sensor Ultrasonik

```
#include <NewPing.h>

//pin arduino yang digunakan
#define trigPin 2
#define echoPin 3

// Tentukan jarak maksimum yang ingin kita ping (dalam sentimeter). Jarak sensor maksimum adalah 40
#define MAX_DISTANCE 400

NewPing sonar = NewPing(trigPin, echoPin, MAX_DISTANCE);

void setup() {
//Baut komunikasi yang disettingkan di serial monitor
```

```
  Serial.begin(9600);
}

void loop() {
  delay(50);
  Serial.print("Jarak Pembacaan = ");
  Serial.print(sonar.ping_cm());  //nilai jarak yang terbaca
  Serial.println(" cm");
  delay(2000);
}
```

Lampiran 6 Program Pengujian Motor Pakan Ikan

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);

void setup() {
  lcd.begin();
  lcd.backlight();
}

void loop() {
  lcd.setCursor (0, 0);
// pindahkan kursor ke baris pertama   lcd.print ( "LCD 20x4" );          // cetak
pesan di baris pertama
   lcd.setCursor (0, 1);
// pindahkan kursor ke baris kedua   lcd.print ( "Alamat I2C: 0x27" );
// cetak pesan di baris kedua   lcd.setCursor (0, 2);            //pindahkan kursor
lcd.print ( "KSA" );          // cetak pesan di baris ketiga
lcd.setCursor   ( 0, 3);            //
```

```
lcd.print ( "LCD I2C" );    // cetak pesan di baris keempat
 }
```

Lampiran 7 Program Pengujian LCD I2C

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);

void setup() {
  lcd.begin();
  lcd.backlight();
}

void loop() {
  lcd.setCursor (0, 0);
// pindahkan kursor ke baris pertama   lcd.print ( "LCD 20x4" );          // cetak
pesan di baris pertama
   lcd.setCursor (0, 1);
// pindahkan kursor ke baris kedua   lcd.print ( "Alamat I2C: 0x27" );
// cetak pesan di baris kedua   lcd.setCursor (0, 2);            //pindahkan kursor
lcd.print ( "KSA" );          // cetak pesan di baris ketiga
lcd.setCursor   ( 0, 3);            //
lcd.print ( "LCD I2C" );    // cetak pesan di baris keempat
 }
```

Lampiran 8 Program Integrasi Sistem

```
#include <LiquidCrystal_I2C.h>
#include "Sodaq_DS3231.h"
#include <Servo.h>
```

```
#include <EEPROM.h>
#include "GravityTDS.h"
#include <Fuzzy.h>
#include <OneWire.h>
#include <DallasTemperature.h>

LiquidCrystal_I2C lcd (0x27, 20, 4);

//Inisiasi Pakan
#define pinServoMakanan          A0
#define waktuBukaServo           2000//milidetik
#define servoBuka                90//derajat
#define servoTutup               180//derajat
#define waktuMakan1              DateTime(0, 0, 1,  9, 0, 0, 0)//jam 9 pagi
#define waktuMakan2              DateTime(0, 0, 1,  15, 0, 0, 0)//jam 3 sore
#define waktuMakan3              DateTime(0, 0, 1,  21, 0, 0, 0)//jam 21 malam
//#define waktuMakan4            DateTime(0, 0, 1,  3, 0, 0, 0)//jam 3 subuh
Servo servoMakanIkan;
byte detikSebelumnya;
char buf[17];
//Inisasi Jarak
const int trigPin = 2;
const int echoPin = 3;
long duration;
int distance;
float realdistance;
//Inisiasi pH
const int phSensorPin  = A2;
float phValue = 0 ;
```

```
// Inisiasi TDS
#define TdsSensorPin A1
GravityTDS gravityTds;
float tdsValue = 0, temp = 25;
//Inisiasi Suhu
#define ONE_WIRE_BUS 4
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
float temperature;
//Inisiasi Aktuator
const int heaterRelayPin = 8;
const int fanRelayPin = 11;
const int valveinPin = 10;
const int valveoutPin = 9;
int buzzerPin = 7;

unsigned long previousMillis = 0; // Waktu terakhir bacaan
const unsigned long interval = 1000;

// Fuzzifikasi
Fuzzy *fuzzy = new Fuzzy();

// FuzzyInput for turbidity
FuzzySet *turbidityLow = new FuzzySet(0, 0, 200, 450);
FuzzySet *turbidityMedium = new FuzzySet(350, 500, 500, 650);
FuzzySet *turbidityHigh = new FuzzySet(550, 800, 1000, 1000);
// FuzzyInput for acidity
FuzzySet *acidityAcid = new FuzzySet(0, 0, 4.5, 6);
FuzzySet *acidityNetral = new FuzzySet(5.5, 7, 7, 8.5);
```

```
FuzzySet *acidityBase = new FuzzySet(8, 10, 14, 14);
// FuzzyOutput for water quality
FuzzySet *cleanQuality = new FuzzySet(0, 0, 3, 4.5);
FuzzySet *turbidQuality = new FuzzySet(3.5, 5, 5, 6.5);
FuzzySet *dirtyQuality = new FuzzySet(5.5, 7, 10, 10);

void setup() {

  servoMakanIkan.attach(pinServoMakanan);
  servoMakanIkan.write(servoTutup);
  rtc.begin();
  //DateTime dt(2024, 3, 6, 21, 6, 0, 4); // set tanggal dan waktu (format): tahun, bulan tanggal, jam, menit, detik, hari (1=minggu, 7=sabtu)
  //rtc.setDateTime(dt);
  Serial.println("Sistem mulai");
  sprintf(buf, "Set waktu 1 = %02d:%02d (%lu)", waktuMakan1.hour(), waktuMakan1.minute(), waktuMakan1.get());
  Serial.println(buf);
  sprintf(buf, "Set waktu 2 = %02d:%02d (%lu)", waktuMakan2.hour(), waktuMakan2.minute(), waktuMakan2.get());
  Serial.println(buf);
  sprintf(buf, "Set waktu 3 = %02d:%02d (%lu)", waktuMakan3.hour(), waktuMakan3.minute(), waktuMakan3.get());
  Serial.println(buf);
  //sprintf(buf, "Set waktu 4 = %02d:%02d (%lu)", waktuMakan4.hour(), waktuMakan4.minute(), waktuMakan4.get());
  //Serial.println(buf);

  Serial.begin(9600);
  randomSeed(analogRead(0));
```

```
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);

gravityTds.setPin(TdsSensorPin);
gravityTds.setAref(5.0);
gravityTds.setAdcRange(1024);
gravityTds.begin();

sensors.begin();

pinMode(heaterRelayPin, OUTPUT);
pinMode(fanRelayPin, OUTPUT);
pinMode(valveinPin, OUTPUT);
pinMode(valveoutPin, OUTPUT);
pinMode(buzzerPin, OUTPUT);

lcd.begin() ;
lcd.backlight();
lcd.setCursor(5, 1);
lcd.print("Welcome to");
delay(1000);
lcd.clear();
lcd.setCursor(2, 2);
lcd.print("Control Aquarium");
delay(1000);
lcd.clear();
lcd.setCursor(3, 1);
lcd.print("System Loading");
for (int a = 0; a <= 19; a++) {
```

```
  lcd.setCursor(a, 2);
  lcd.print(".");
  delay(100);
 }
 lcd.clear();


// FuzzyInput for turbidity
FuzzyInput *turbidityLevel = new FuzzyInput(1);
turbidityLevel->addFuzzySet(turbidityLow);
turbidityLevel->addFuzzySet(turbidityMedium);
turbidityLevel->addFuzzySet(turbidityHigh);

fuzzy->addFuzzyInput(turbidityLevel);

// FuzzyInput for acidity
FuzzyInput *acidityLevel = new FuzzyInput(2);
acidityLevel->addFuzzySet(acidityAcid);
acidityLevel->addFuzzySet(acidityNetral);
acidityLevel->addFuzzySet(acidityBase);

fuzzy->addFuzzyInput(acidityLevel);

// FuzzyOutput for water quality
FuzzyOutput *waterQuality = new FuzzyOutput(1);
waterQuality->addFuzzySet(cleanQuality);
waterQuality->addFuzzySet(turbidQuality);
waterQuality->addFuzzySet(dirtyQuality);
```

```
 fuzzy->addFuzzyOutput(waterQuality);

 // Building FuzzyRule1
 FuzzyRuleAntecedent *rule1Antecedent = new FuzzyRuleAntecedent();
 rule1Antecedent->joinWithAND(turbidityLow, acidityAcid);
 FuzzyRuleConsequent *rule1Consequent = new FuzzyRuleConsequent();
 rule1Consequent->addOutput(turbidQuality);
 FuzzyRule *fuzzyRule1 = new FuzzyRule(1, rule1Antecedent,
rule1Consequent);
 fuzzy->addFuzzyRule(fuzzyRule1);

 // Building FuzzyRule2
 FuzzyRuleAntecedent *rule2Antecedent = new FuzzyRuleAntecedent();
 rule2Antecedent->joinWithAND(turbidityMedium, acidityAcid);
 FuzzyRuleConsequent *rule2Consequent = new FuzzyRuleConsequent();
 rule2Consequent->addOutput(dirtyQuality);
 FuzzyRule *fuzzyRule2 = new FuzzyRule(2, rule2Antecedent,
rule2Consequent);
 fuzzy->addFuzzyRule(fuzzyRule2);

 // Building FuzzyRule3
 FuzzyRuleAntecedent *rule3Antecedent = new FuzzyRuleAntecedent();
 rule3Antecedent->joinWithAND(turbidityHigh, acidityAcid);
 FuzzyRuleConsequent *rule3Consequent = new FuzzyRuleConsequent();
 rule3Consequent->addOutput(dirtyQuality);
 FuzzyRule *fuzzyRule3 = new FuzzyRule(3, rule3Antecedent,
rule3Consequent);
 fuzzy->addFuzzyRule(fuzzyRule3);
```

```
 // Building FuzzyRule4
 FuzzyRuleAntecedent *rule4Antecedent = new FuzzyRuleAntecedent();
 rule4Antecedent->joinWithAND(turbidityLow, acidityNetral);
 FuzzyRuleConsequent *rule4Consequent = new FuzzyRuleConsequent();
 rule4Consequent->addOutput(cleanQuality);
 FuzzyRule *fuzzyRule4 = new FuzzyRule(4, rule4Antecedent,
rule4Consequent);
 fuzzy->addFuzzyRule(fuzzyRule4);

 // Building FuzzyRule5
 FuzzyRuleAntecedent *rule5Antecedent = new FuzzyRuleAntecedent();
 rule5Antecedent->joinWithAND(turbidityMedium, acidityNetral);
 FuzzyRuleConsequent *rule5Consequent = new FuzzyRuleConsequent();
 rule5Consequent->addOutput(turbidQuality);
 FuzzyRule *fuzzyRule5 = new FuzzyRule(5, rule5Antecedent,
rule5Consequent);
 fuzzy->addFuzzyRule(fuzzyRule5);

 // Building FuzzyRule6
 FuzzyRuleAntecedent *rule6Antecedent = new FuzzyRuleAntecedent();
 rule6Antecedent->joinWithAND(turbidityHigh, acidityNetral);
 FuzzyRuleConsequent *rule6Consequent = new FuzzyRuleConsequent();
 rule6Consequent->addOutput(dirtyQuality);
 FuzzyRule *fuzzyRule6 = new FuzzyRule(6, rule6Antecedent,
rule6Consequent);
 fuzzy->addFuzzyRule(fuzzyRule6);

 // Building FuzzyRule7
 FuzzyRuleAntecedent *rule7Antecedent = new FuzzyRuleAntecedent();
```

```
 rule7Antecedent->joinWithAND(turbidityLow, acidityBase);
 FuzzyRuleConsequent *rule7Consequent = new FuzzyRuleConsequent();
 rule7Consequent->addOutput(turbidQuality);
 FuzzyRule *fuzzyRule7 = new FuzzyRule(7, rule7Antecedent,
rule7Consequent);
 fuzzy->addFuzzyRule(fuzzyRule7);

 // Building FuzzyRule8
 FuzzyRuleAntecedent *rule8Antecedent = new FuzzyRuleAntecedent();
 rule8Antecedent->joinWithAND(turbidityMedium, acidityBase);
 FuzzyRuleConsequent *rule8Consequent = new FuzzyRuleConsequent();
 rule8Consequent->addOutput(dirtyQuality);
 FuzzyRule *fuzzyRule8 = new FuzzyRule(8, rule8Antecedent,
rule8Consequent);
 fuzzy->addFuzzyRule(fuzzyRule8);

 // Building FuzzyRule9
 FuzzyRuleAntecedent *rule9Antecedent = new FuzzyRuleAntecedent();
 rule9Antecedent->joinWithAND(turbidityHigh, acidityBase);
 FuzzyRuleConsequent *rule9Consequent = new FuzzyRuleConsequent();
 rule9Consequent->addOutput(dirtyQuality);
 FuzzyRule *fuzzyRule9 = new FuzzyRule(9, rule9Antecedent,
rule9Consequent);
 fuzzy->addFuzzyRule(fuzzyRule9);

}

void getdatasuhu() {
 //Nilai Suhu
```

```
sensors.requestTemperatures();
temperature = sensors.getTempCByIndex(0);
 Serial.print("Suhu: ");
 Serial.println(sensors.getTempCByIndex(0));
 //delay(1000);
 lcd.setCursor(0, 1);
 lcd.print("Suhu:");
 lcd.setCursor(5, 1);
 lcd.print(temperature,1);
 // Atur cooler berdasarkan suhu
if (temperature > 30.00) {
 digitalWrite(fanRelayPin, LOW); // Nyalakan kipas saat suhu < 35
 }
else if (temperature <= 30.00) {
 digitalWrite(fanRelayPin, HIGH); // Matikan kipas saat suhu = 35
 }
 // Atur heater berdasarkan suhu
if (temperature < 24.00) {
  digitalWrite(heaterRelayPin, LOW); // Nyalakan heater saat suhu < 24
 }
else if (temperature >= 24.00) {
  digitalWrite(heaterRelayPin, HIGH); // Matikan heater saat suhu > 30
 }
}

void getdatadistance() {
//Nilai Jarak
 int duration, distance, realdistance;
 digitalWrite(trigPin,LOW);
```

```
 delayMicroseconds(2);
 digitalWrite(trigPin,HIGH);
 delayMicroseconds(10);
 digitalWrite(trigPin,LOW);
 duration = pulseIn(echoPin,HIGH);
 distance = duration*0.034/2;
 realdistance = -((distance-45));
 Serial.print("Jarak Pembacaan = ");
 Serial.print(realdistance);  //nilai jarak yang terbaca
 Serial.println(" cm");
 //delay(1000);
 lcd.setCursor(0, 2);
 lcd.print("Tinggi:");
 lcd.setCursor(7, 2);
 lcd.print(realdistance);
 if(realdistance <= 15)
  {
  digitalWrite(valveoutPin, HIGH);
  digitalWrite(valveinPin, LOW); }
  else if (realdistance >= 20) {
  digitalWrite(valveinPin, HIGH);
  }
}

void getdataph() {
//Nilai pH
int nilaiPengukuranPh = analogRead(phSensorPin);
   double TeganganPh = 5 / 1024.0 * nilaiPengukuranPh;
   phValue = 7.00 + ((3.03 - TeganganPh) / 0.12);
```

```
    //PH7 = 3,03 , PH4 = 3,37
    //Po = 7.00 + ((teganganPh7 - TeganganPh) / PhStep);
  Serial.print("Nilai PH cairan: ");
  Serial.println(phValue, 2);
  //delay(1000);
  lcd.setCursor(11, 2);
  lcd.print("PH:");
  lcd.setCursor(14, 2);
  lcd.print(phValue, 2);
}

void getdatatds() {
  //Nilai TDS
  gravityTds.setTemperature(temp);
  gravityTds.update();
  tdsValue = gravityTds.getTdsValue();
  Serial.print(tdsValue,0);
  Serial.println("ppm");
  // delay(1000);
  lcd.setCursor(11, 1);
  lcd.print("TDS:");
  lcd.setCursor(15, 1);
  lcd.print(tdsValue, 1);
}

void loop() {

  getdatatds();
  getdataph();
```

```
 getdatasuhu();
 getdatadistance();

 unsigned long currentMillis = millis();
 // Periksa apakah sudah mencapai interval
 if (currentMillis - previousMillis >= interval) {
previousMillis = currentMillis; // Update waktu terakhir bacaan
}

DateTime now = rtc.now();
 if (detikSebelumnya != now.second())
 {
   sprintf(buf, "%02d:%02d:%02d", now.hour(), now.minute(), now.second());
   lcd.setCursor(0, 0);
   lcd.print(buf);
   Serial.print(buf);

   detikSebelumnya = now.second();

   uint32_t epoch = now.get() % 86400;//hanya jam menit detik

   if ((epoch == waktuMakan1.get()) ||
      (epoch == waktuMakan3.get()) ||
      (epoch == waktuMakan3.get()) ||
      (epoch == waktuMakan4.get()))
   {
     char buf[17];
     sprintf(buf, "Time: %02d:%02d", now.hour(), now.minute());
     lcd.setCursor(9, 0);
```

```
    lcd.print(buf);
    Serial.println(buf);
    delay(1000);

    servoMakanIkan.write(servoBuka);
    delay(waktuBukaServo);
    servoMakanIkan.write(servoTutup);

  }
 }

fuzzy->setInput(1, tdsValue);
fuzzy->setInput(2, phValue);

 fuzzy->fuzzify();
 Serial.print("Turbidity Level: Low-> ");
 Serial.print(turbidityLow->getPertinence());
 Serial.print(", Medium-> ");
 Serial.print(turbidityMedium->getPertinence());
 Serial.print(", High-> ");
 Serial.println(turbidityHigh->getPertinence());
 Serial.print("Acidity Level: Acid-> ");
 Serial.print(acidityAcid->getPertinence());
 Serial.print(", Netral-> ");
 Serial.print(acidityNetral->getPertinence());
 Serial.print(", Base-> ");
 Serial.print(acidityBase->getPertinence());

 float waterQualityOutput = fuzzy->defuzzify(1);
```

```
 Serial.println("Output: ");
 Serial.print("Water Quality: Clean-> ");
 Serial.print(cleanQuality->getPertinence());
 Serial.print(", Turbid-> ");
 Serial.print(turbidQuality->getPertinence());
 Serial.print(", Dirty-> ");
 Serial.print(dirtyQuality->getPertinence());
 Serial.print("Result: ");
 Serial.println(waterQualityOutput);
 String waterQuality;
 if (waterQualityOutput >= 0 && waterQualityOutput <= 4) {
 waterQuality = "Jernih";
 } else if (waterQualityOutput >= 4 && waterQualityOutput <= 6) {
 waterQuality = "Keruh";
 } else if (waterQualityOutput >= 6 && waterQualityOutput <= 10) {
 waterQuality = "Kotor";
 }
 Serial.print("Kualitas Air: ");
 Serial.println(waterQuality);
 lcd.setCursor(0, 3);
 lcd.print("Kualitas: ");
 lcd.print(waterQuality);
if (waterQualityOutput >= 6 && waterQualityOutput <= 10) {
  tone(buzzerPin, 1000, 1000);
  digitalWrite(valveoutPin, LOW);
  delay(9000);
  }
else {
```

```
 digitalWrite(valveoutPin, HIGH);
 noTone (buzzerPin);
 }

}
```

Lampiran 9 Percobaan Jumlah Pakan

Waktu (S)	Jumlah Pakan (Biji)		
	Percobaan 1	Percobaan 2	Percobaan 3
1	236	192	217
2	578	603	611

Lampiran 10 Dokumentasi Penelitian