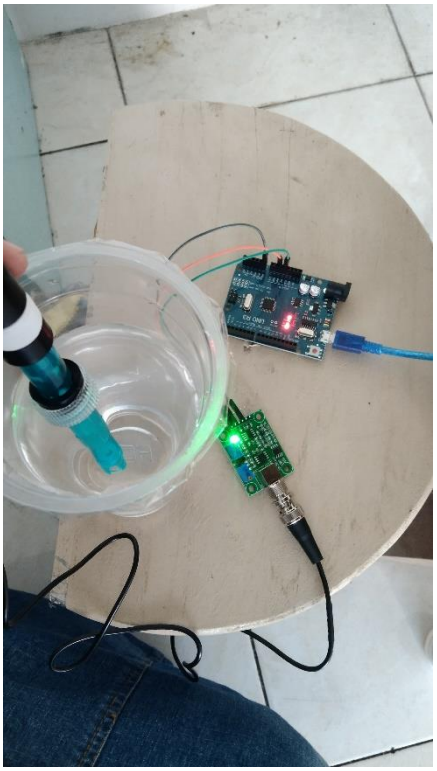
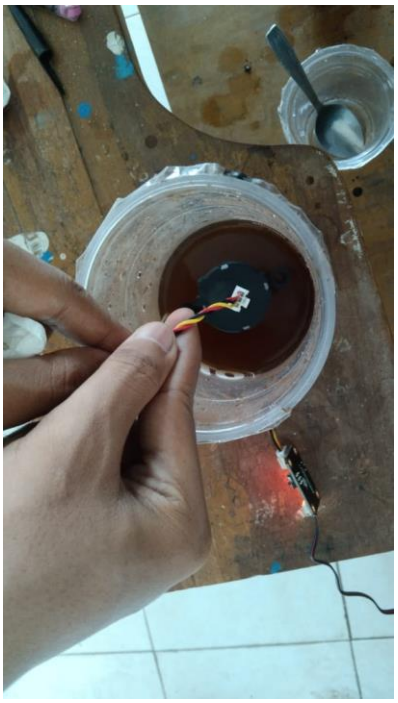
LAMPIRAN

Lampiran – 2 Dokumentasi Alat














pH Down
Menurunkan Nilai pH
HNO3 10%



LAMPIRAN

Lampiran – 1 Program

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

// === Pin Konstanta ===
#define PH_PIN A0
#define TDS_PIN A1
#define TURBIDITY_PIN A2
#define TRIG_PIN 2
#define ECHO_PIN 3

#define RELAY_AIR 5         // Pompa air biasa
#define RELAY_PH_DOWN 4     // Pompa penurun pH

#define ENA 6
#define IN1 7
#define IN2 8

// === Batas Logika ===
float PH_UPPER = 5.0;
float PH_LOWER = 3.0;
float TANK_FULL_CM = 10.0;

unsigned long lastLCDSwitch = 0;
int lcdPage = 0;

bool phTujuanTercapai = false;

// === Motor Control ===
void runMotor() {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  analogWrite(ENA, 180);
}
```

```
void stopMotor() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  analogWrite(ENA, 0);
}

void setup() {
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  lcd.print("Initializing...");

  pinMode(PH_PIN, INPUT);
  pinMode(TDS_PIN, INPUT);
  pinMode(TURBIDITY_PIN, INPUT);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);

  pinMode(RELAY_AIR, OUTPUT);
  pinMode(RELAY_PH_DOWN, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(ENA, OUTPUT);

  digitalWrite(RELAY_AIR, LOW);
  digitalWrite(RELAY_PH_DOWN, LOW);
  stopMotor();

  delay(2000);
  lcd.clear();
}

void loop() {
  float pH = readPH();
  float tds = readTDS();
  float turb = readTurbidity();
  float jarak = readUltrasonicAvg();

  bool tankNotFull = (jarak > TANK_FULL_CM);
  bool phTooLow = (pH < PH_LOWER);

  bool pompa1Aktif = false;
  bool pompa2Aktif = false;
```

```
// === Logika Pompa 2 ===
  if (pH > PH_UPPER && !phTujuanTercapai && tankNotFull) {
    pompa2Aktif = true;
  }

  if (pH <= PH_UPPER && !phTujuanTercapai) {
    phTujuanTercapai = true;
  }

  if (pH > PH_UPPER + 0.2) {
    phTujuanTercapai = false;
  }

  // === Logika Pompa 1 ===
  if (tankNotFull) {
    pompa1Aktif = true;
  }

  digitalWrite(RELAY_PH_DOWN, pompa2Aktif ? HIGH : LOW);
  digitalWrite(RELAY_AIR, pompa1Aktif ? HIGH : LOW);

  if (pompa1Aktif || pompa2Aktif) {
    runMotor();
  } else {
    stopMotor();
  }

  // === Tampilan LCD ===
  if (millis() - lastLCDSwitch > 3000) {
    lastLCDSwitch = millis();
    lcdPage = (lcdPage + 1) % 3;
    lcd.clear();
  }

  if (lcdPage == 0) {
    lcd.setCursor(0, 0);
    lcd.print("pH: ");
    lcd.print(pH, 1);
    lcd.print(" Lv:");
    lcd.print(tankNotFull ? "Low " : "Full");

 lcd.setCursor(0, 1);
    if (pompa2Aktif)
```

```
lcd.print("Add pH Down");
    else if (pompa1Aktif)
      lcd.print("Add Water");
    else
      lcd.print("Stable");
  }

  else if (lcdPage == 1) {
    lcd.setCursor(0, 0);
    lcd.print("TDS: ");
    lcd.print(tds, 0);
    lcd.print("ppm");

    lcd.setCursor(0, 1);
    lcd.print("Turb: ");
    lcd.print(turb, 1);
    lcd.print(" NTU");
  }

  else if (lcdPage == 2) {
    lcd.setCursor(0, 0);
    lcd.print("Water Quality:");

    lcd.setCursor(0, 1);
    if (pH <= PH_UPPER) {
      lcd.print("LAYAK");
    } else {
      lcd.print("TIDAK LAYAK");
    }
  }

  // === Debug Serial Monitor ===
  Serial.print("pH: "); Serial.print(pH);
  Serial.print(" | TDS: "); Serial.print(tds);
  Serial.print(" ppm | Turb: "); Serial.print(turb);
  Serial.print(" NTU | Jarak: "); Serial.print(jarak);
Serial.println(" cm");

  delay(200);
}
// === Fungsi Sensor ===
float readPH() {
  int val = analogRead(PH_PIN);
  float volt = val * (5.0 / 1023.0);
```

```
float ph = 7 + ((3.3 - volt) * 3.5); // Kalibrasi bisa
disesuaikan
  return ph;
}

float readTDS() {
  int val = analogRead(TDS_PIN);
  float volt = val * (5.0 / 1023.0);
  return volt * 500.0;
}

float readTurbidity() {
  int val = analogRead(TURBIDITY_PIN);
  float volt = val * (5.0 / 1023.0);
  return (volt < 2.5) ? (2.5 - volt) * 3.6 : 0;
}

float readUltrasonic() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  long dur = pulseIn(ECHO_PIN, HIGH);
  float cm = dur * 0.0343 / 2;
  return cm;
}

float readUltrasonicAvg() {
  float total = 0;
  for (int i = 0; i < 5; i++) {
    total += readUltrasonic();
    delay(50);
  }
  return total / 5.0;
}
```

1. **Kalibrasi sensor TDS DFRobot**

Referensi sensor: Gravity Analog TDS Sensor v1.0

Rumus konversi: DFRobot menggunakan rumus:

```
TDS = (133.42 * voltage^3 - 255.86 * voltage^2 + 857.39 * voltage) * 0.5
#define TDS_PIN A1
float analogVoltage, tdsValue;
void setup() {
  Serial.begin(9600);
  pinMode(TDS_PIN, INPUT);
}
void loop() {
  int analogValue = analogRead(TDS_PIN);
  analogVoltage = analogValue * (5.0 / 1024.0);  // Tegangan ADC 10-bit
  // Rumus dari DFRobot
  tdsValue = (133.42 * pow(analogVoltage, 3) - 255.86 *
pow(analogVoltage, 2) + 857.39 * analogVoltage) * 0.5;
  Serial.print("Analog: "); Serial.print(analogValue);
  Serial.print(" | Tegangan: "); Serial.print(analogVoltage, 2); Serial.print("
V");
  Serial.print(" | TDS: "); Serial.print(tdsValue); Serial.println(" ppm");
  delay(1000);
}
```

2. Kalibrasi Sensor Turbidity DFRobot

Referensi sensor: Gravity Analog Turbidity Sensor v1.0

Catatan: DFRobot menggunakan tegangan analog dan interpolasi linear untuk

NTU. Rumus dari datasheet:

- Semakin tinggi NTU (keruh), semakin rendah tegangan.
- Contoh kalibrasi (gunakan air jernih dan keruh sebagai acuan):

```
#define TURB_PIN A0
float voltage;
float turbidity;
void setup() {
  Serial.begin(9600);
  pinMode(TURB_PIN, INPUT);
}
void loop() {
  int sensorValue = analogRead(TURB_PIN);
  voltage = sensorValue * (5.0 / 1024.0);  // Tegangan sensor
  // Rumus konversi dari DFRobot (perkiraan)
  turbidity = -1120.4 * voltage * voltage + 5742.3 * voltage - 4352.9;
  Serial.print("Analog: "); Serial.print(sensorValue);
  Serial.print(" | Tegangan: "); Serial.print(voltage, 2); Serial.print(" V");
  Serial.print(" | Turbidity: "); Serial.print(turbidity); Serial.println(" NTU");
  delay(1000);
}
```

- Gunakan **air murni (TDS = 0 ppm)** dan larutan **dengan nilai ppm diketahui** untuk menguji akurasi sensor TDS.

 Untuk turbidity, gunakan **air jernih** (sekitar 0 NTU) dan air keruh dengan **standar laboratorium** atau **susu/larutan sabun** untuk perbandingan nilai.

 Anda bisa menambahkan **offset kalibrasi manual** jika nilai terlalu jauh dari nilai seharusnya.

3. Kalibrasi Sensor pH

```
#include <EEPROM.h>
#define PH_PIN A0            // Pin sensor pH
#define EEPROM_ADDR 0        // Alamat EEPROM untuk simpan offset
#define VREF 5.0             // Tegangan referensi (V)
#define ADC_RES 1024.0       // Resolusi ADC 10-bit
#define CALIBRATION_BUTTON 7   // Pin tombol kalibrasi pH 7
float voltage, phValue;
float phOffset = 0.0;
void setup() {
  Serial.begin(9600);
  pinMode(PH_PIN, INPUT);
  pinMode(CALIBRATION_BUTTON, INPUT_PULLUP);
  // Ambil offset dari EEPROM
  EEPROM.get(EEPROM_ADDR, phOffset);
  if (isnan(phOffset)) phOffset = 0.0;
  Serial.println("DFRobot Gravity: Analog pH Sensor Calibration");
  Serial.print("Offset pH saat ini: "); Serial.println(phOffset, 3);
  Serial.println("Tekan tombol untuk kalibrasi pH 7.00");
}
void loop() {
  int adcValue = analogRead(PH_PIN);
  voltage = adcValue * (VREF / ADC_RES);
  phValue = 3.5 * voltage + phOffset;
```

```
Serial.print("Tegangan: "); Serial.print(voltage, 3); Serial.print(" V");
  Serial.print(" | pH: "); Serial.println(phValue, 2);
  // Jika tombol ditekan, lakukan kalibrasi
  if (digitalRead(CALIBRATION_BUTTON) == LOW) {
    calibratePH7(voltage);
    delay(2000);  // Debounce
  }
  delay(1000);
}
void calibratePH7(float voltageInput) {
  // Saat larutan buffer pH 7.00 digunakan
  float newOffset = 7.00 - (3.5 * voltageInput);
  phOffset = newOffset;
  EEPROM.put(EEPROM_ADDR, phOffset);
  Serial.println("Kalibrasi selesai!");
  Serial.print("Offset baru: "); Serial.println(phOffset, 3);
}
```

- **Tombol kalibrasi**: Sambungkan ke pin 7 dan GND. Tekan saat probe berada di larutan pH 7.00.
- **EEPROM**: Nilai offset disimpan sehingga tidak hilang saat Arduino di-reset.
- **Formula default DFRobot**:

```
pH = 3.5 * tegangan + offset
```

KARTU MONITORING BIMBINGAN

MAHASISWA PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

PROPOSAL

Mahasiswa : NISMA	**Pembimbing I** : Alauddin Y, S.T., M.Kom.
NIM : 218180064	**Pembimbing II** : Ashadi Amir, S.T., M.T.
Judul Skripsi : SISTEM PENGONTROLAN PENGISIAN DAN PENDISTRIBUSIAN AIR GAMBUT PADA PETERNAKAN AYAM BROILER BERBASIS MIKROKONTROLER	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 1. Gabar konstruksi alat x 2. Flowchart	21/12/2023	Konsultasi 1 1. Rumusan Masalah 2. Tujuan penelitian	
Konsultasi 2 1. Keterangan yang tidak di gambar cantumkan	2/1/2024	Konsultasi 2 Tinjauan pustaka	
Konsultasi 3 2. Flowchart alat	4/1/2024	Konsultasi 3 Metode Penelitian	
Konsultasi 4 Rangkaian komponen yg digunakan.	7/12/23	Konsultasi 4 Teknik Pengumpulan Data	
Konsultasi 5 Acc	20/1/2024	Konsultasi 5 ACC	

Lanjut ke halaman sebelah...

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton berwarna hijau muda dan dicetak timbal balik

Lanjutan ...

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 6		Konsultasi 6	
Konsultasi 7		Konsultasi 7	
Konsultasi 8		Konsultasi 8	
Konsultasi 9		Konsultasi 9	
Konsultasi 10		Konsultasi 10	

Parepare, 16 Juni 2025

Mengetahui
Ketua Program Studi

Ashadi Amir, ST., MT.
NBM. 1278 015

Mahasiswa

NISMA
NIM. ***218180064***

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton berwarna hijau muda dan dicetak timbal balik