DAFTAR LAMPIRAN

Lampiran- 1 Program alat

```
#include<Arduino.h>
#include<SoftwareSerial.h>

// Define pins for sensor 1
#define TRIG_PIN_1 9
#define ECHO_PIN_1 10

// Define pins for sensor 2
#define TRIG_PIN_2 11
#define ECHO_PIN_2 12

// Define pins for relay channels
#define RELAY_PIN_1 4
#define RELAY_PIN_2 5

#define SensorPin1 A0 // pH meter Analog output to Arduino Analog
Input 0
#define SensorPin2 A1 // pH meter Analog output to Arduino Analog
Input 1
#define SensorPin3 A5 // pH meter Analog output to Arduino Analog
Input 5

floatcalibration_value1 = 21.34 + 2.5; // Calibration value for
Sensor 1
floatcalibration_value2 = 21.34 + 3.1; // Calibration value for
Sensor 2
floatcalibration_value3 = 21.34 + 3.8; // Calibration value for
Sensor 3

#define analogInPinTDS1 A3 // Analog input pin untuk tds 1
#define analogInPinTDS2 A2 // Analog input pin untuk tds 2
#define analogInPinTDS3 A4 // Analog input pin untuk tds 3


// variable
// int sensorValue0; // adc value
floattds_calibration_value0 = 1203.08;
floattds_calibration_value1 = 1203.08;
floattds_calibration_value2 = 1203.08;
```

```
SoftwareSerialmySerial(2, 3); // RX, TX , Software serial from ESP-
01 to Arduino

intdetikGerbang = 0;

boolbukaGerbang = false;
booltutupGerbang = false;

StringstatusGerbang = "idle";

floattdsReading(intsensorPin, floatcalibration_value)
{
  intsensorValue = analogRead(sensorPin);
  floatoutputTds = (0.3417 * sensorValue) + calibration_value;
  returnoutputTds;
}

floatprocessPHsensor(intpin, floatcalibration_value)
{
  intbuf[10];
  inttemp;
  unsignedlongintavgValue;

  // Read and smooth the sensor data
  for (inti = 0; i<10; i++)
  {
    buf[i] = analogRead(pin);
    delay(10);
  }

  // Sort the values
  for (inti = 0; i<9; i++)
  {
    for (intj = i + 1; j<10; j++)
    {
      if (buf[i] >buf[j])
      {
        temp = buf[i];
        buf[i] = buf[j];
        buf[j] = temp;
      }
    }
  }

  // Calculate the average of the middle 6 values
```

```
  avgValue = 0;
  for (inti = 2; i<8; i++)
    avgValue += buf[i];

  // Convert the average value to pH
  floatphValue = (float)avgValue * 5.0 / 1024 / 6;
  phValue = -5.70 * phValue + calibration_value;

  returnphValue;
}

longmeasureDistance(inttrigPin, intechoPin)
{
  // Clear the trigger
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  // Send a 10 microsecond pulse to the trigger
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Read the pulse duration
  returnpulseIn(echoPin, HIGH);
}

voidsetup()
{
  Serial.begin(115200);
  mySerial.begin(115200);

  // built in led
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_BUILTIN, LOW);

  // initialize ph sensor
  pinMode(SensorPin1, INPUT);
  pinMode(SensorPin2, INPUT);
  pinMode(SensorPin3, INPUT);

  // Initialize sensor 1
  pinMode(TRIG_PIN_1, OUTPUT);
  pinMode(ECHO_PIN_1, INPUT);

  // Initialize sensor 2
```

```
  pinMode(TRIG_PIN_2, OUTPUT);
  pinMode(ECHO_PIN_2, INPUT);

  // Initialize relay pins
  pinMode(RELAY_PIN_1, OUTPUT);
  pinMode(RELAY_PIN_2, OUTPUT);

  // Ensure relays are off at the start
  digitalWrite(RELAY_PIN_1, LOW);
  digitalWrite(RELAY_PIN_2, LOW);

  // delay(1000);
}

voidloop()
{
  StringIncomingStr = "";
  boolflag = false;

  // Check if data is available from ESP-01
  while (mySerial.available())
  {
    IncomingStr=mySerial.readString();
    flag = true;
  }

  Serial.println("Ini flag : "+String(flag));

  if (flag)
  {
    Stringresponse = IncomingStr;
    // Serial.println(response); // Print to Serial monitor for
debugging
    // mySerial.print("aran"); // Send response back to ESP-01
    // mySerial.print(i);
    // mySerial.print ("\n");
    intcommaIndex1 = response.indexOf(',');
    Stringwifi = response.substring(0, commaIndex1);
    Stringgerbang = response.substring(commaIndex1 + 1);
    Serial.println(wifi);
    Serial.println(gerbang);

    if (wifi=="Wifi connected")
    {
      Serial.println("Wifi Connected");
```

```
      digitalWrite(LED_BUILTIN, HIGH);
    }
    else
    {
      Serial.println("Wifi Not Connected");
      digitalWrite(LED_BUILTIN, LOW);
    }

     // Assuming `gerbang` is the received command
    if (gerbang=="buka"&&statusGerbang!="buka") {
        bukaGerbang = true;
    } elseif (gerbang=="tutup"&&statusGerbang!="tutup") {
        tutupGerbang = true;
    }

    // Handle the "buka" command
    if (bukaGerbang == true) {
        if (detikGerbang<2) { // Open for 5 seconds
            detikGerbang++;
            // digitalWrite(RELAY_PIN_1, LOW); // Ensure both relays
are off
            // digitalWrite(RELAY_PIN_2, HIGH);
            delay(2500);
            digitalWrite(RELAY_PIN_1, LOW); // Open the gate
            digitalWrite(RELAY_PIN_2, HIGH);  // Ensure the other
relay is off
        } else { // After 5 seconds
            bukaGerbang = false;
            statusGerbang="buka"; // Update the status
            detikGerbang = 0;
            digitalWrite(RELAY_PIN_1, LOW); // Close the gate
            digitalWrite(RELAY_PIN_2, LOW); // Ensure both relays
are off
        }
    }

    // Handle the "tutup" command
    if (tutupGerbang == true) {
        if (detikGerbang<2) { // Close for 5 seconds
            detikGerbang++;
            // digitalWrite(RELAY_PIN_1, HIGH); // Ensure both
relays are off
            // digitalWrite(RELAY_PIN_2, HIGH);
            delay(2500);
```

```
            digitalWrite(RELAY_PIN_1, HIGH); // Ensure the other
relay is off
            digitalWrite(RELAY_PIN_2, LOW); // Close the gate
        } else { // After 5 seconds
            tutupGerbang = false;
            statusGerbang="tutup"; // Update the status
            detikGerbang = 0;
            digitalWrite(RELAY_PIN_1, LOW); // Ensure both relays
are off
            digitalWrite(RELAY_PIN_2, LOW); // Ensure both relays
are off
        }
    }

    // Optionally, print the status
    Serial.println("Gate Status: "+statusGerbang);
  }

  // Measure distance for sensor 1
  longduration1 = measureDistance(TRIG_PIN_1, ECHO_PIN_1);
  floatdistance1 = (duration1 / 2.0) * 0.0344;

  // Measure distance for sensor 2
  longduration2 = measureDistance(TRIG_PIN_2, ECHO_PIN_2);
  floatdistance2 = (duration2 / 2.0) * 0.0344;

  floatphValue1 = processPHsensor(SensorPin1, calibration_value1);
  floatphValue2 = processPHsensor(SensorPin2, calibration_value2);
  floatphValue3 = processPHsensor(SensorPin3, calibration_value3);
  // float phValue2 = processPHsensor(SensorPin2,
calibration_value2);

  //
  floattdsValue1 = tdsReading(analogInPinTDS1,
tds_calibration_value0);
  floattdsValue2 = tdsReading(analogInPinTDS2,
tds_calibration_value1);
  floattdsValue3 = tdsReading(analogInPinTDS3,
tds_calibration_value2);

  // Print the pH values to the serial monitor

  Serial.print("Distance1: ");
  Serial.print(distance1);
  Serial.print("cm ,Distance2: ");
```

```
Serial.print(distance2);
Serial.println("cm ");

Serial.print("pH Value 1: ");
Serial.println(phValue1);

Serial.print("pH Value 2: ");
Serial.println(phValue2);

Serial.print("pH Value 3: ");
Serial.println(phValue3);

Serial.print("TDS Value 1: ");
Serial.println(tdsValue1);

Serial.print("TDS Value 2: ");
Serial.println(tdsValue2);

Serial.print("TDS Value 3: ");
Serial.println(tdsValue3);

// if (distance1 <= 30)
// {
//   digitalWrite(RELAY_PIN_1, HIGH);
//   digitalWrite(RELAY_PIN_2, LOW);
// }
// else if (distance1 >= 60)
// {
//   digitalWrite(RELAY_PIN_2, HIGH);
//   digitalWrite(RELAY_PIN_1, LOW);
// }
// else
// {
//   digitalWrite(RELAY_PIN_1, LOW);
//   digitalWrite(RELAY_PIN_2, LOW);
// }

mySerial.print(statusGerbang);
mySerial.print(",");
mySerial.print(distance1);
mySerial.print(",");
mySerial.print(distance2);
mySerial.print(",");
mySerial.print(phValue1);
mySerial.print(",");
```

```
  mySerial.print(phValue2);
  mySerial.print(",");
  mySerial.print(phValue3);
  mySerial.print(",");
  mySerial.print(tdsValue1);
  mySerial.print(",");
  mySerial.print(tdsValue2);
  mySerial.print(",");
  mySerial.print(tdsValue3);
  mySerial.print("\n");

  delay(2000);
}
```

Lampiran- 2 Alat penelitian

Lampiran- 3 Datasheet

1. sensor tds

SENSOR KONDUKTIVITAS / TDS / KADAR GARAM

Tipe Aplikasi :

- Sensor konduktivitas (conductivity sensor)
- Sensor TDS (total dissolve solid)
- Sensor kadar garam (salinity sensor)

Spesifikasi :

- Bekerja pada tegangan DC 5 Volt
- Support arduino dan mikrokontroller lainnya
- Koefisien linearitas data konduktivitas sebesar 0.9639
- Koefisien linearitas data TDS sebesar 0.983
- Memiliki sensitivitas pada bahan yang bersifat konduktif
- Kedalaman cairan pada saat pengukuran sebesar 5.5 cm dari ujung sensor
- Rumus persamaan umum konversi data konduktivitas **y = 0.2142x + 494.93, dimana : x = nilai ADC, dan y=konduktivitas**
- Rumus persamaan umum konversi data TDS **y= 0.3417x + 281.08, dimana : x = nilai ADC, dan y=TDS**

SENSOR KONDUKTIVITAS / TDS / KADAR GARAM

Pin	Deskripsi
5V	5V arduino
Gnd	GND arduino
Output	Output ke pin A0 arduino

Tabel 1. Pin Sensor

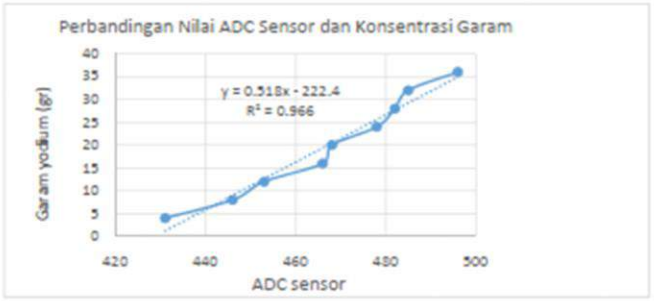
Sensor Konduktivitas / TDS / Kadar Garam memiliki desain yang kompak. Probe sensornya berbahan stik stainless yang berfungsi sebagai penerima data dari bahan yang diuji. Sensor ini dapat langsung disambungkan dengan pin analog arduino maupun pin analog mikrokontroller lainnya, tanpa harus memakai modul penguat tambahan.

KARAKTERISTIK

SENSOR KONDUKTIVITAS / TDS / KADAR GARAM

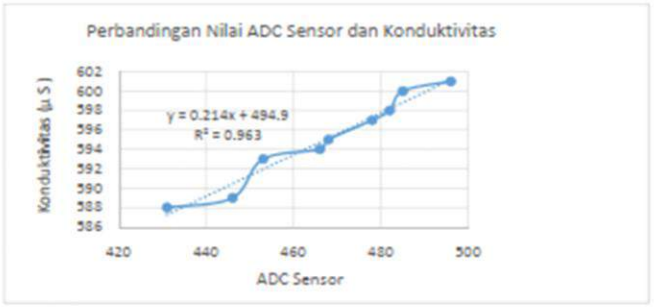
Parameter	Simbol	Min	Max	Units
Tegangan masukan	Vcc		5.0	V
Tegangan operasional	Vcc	3.0	4.7	V
Tegangan keluaran	ADC	0	1023	ADC
Respon waktu	t	0.1	0.3	s
Sensitivitas	Vcc	0.1	0.5	V

Tabel 2. Karakteristik Sensor

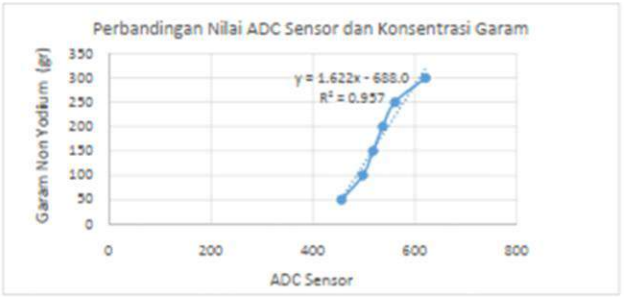


Grafik 3. Karakteristik nilai ADC sensor terhadap konsentrasi garam (uji konduktivitas)

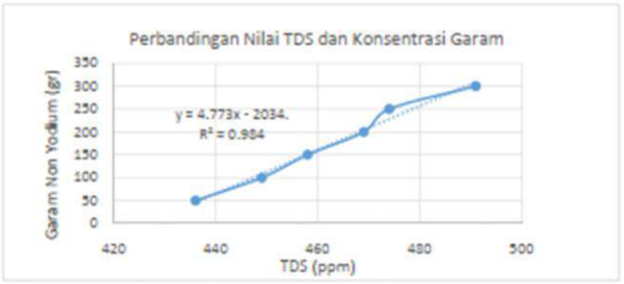
Grafik 4. Karakteristik nilai konduktivitas sensor kalibrator conductivity meter terhadap konsentrasi garam (uji konduktivitas)



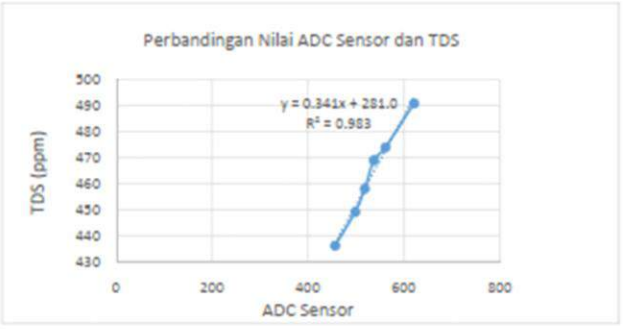
Grafik 5. Karakteristik nilai ADC sensor terhadap nilai konduktivitas sensor kalibrator conductivity meter (uji konduktivitas)



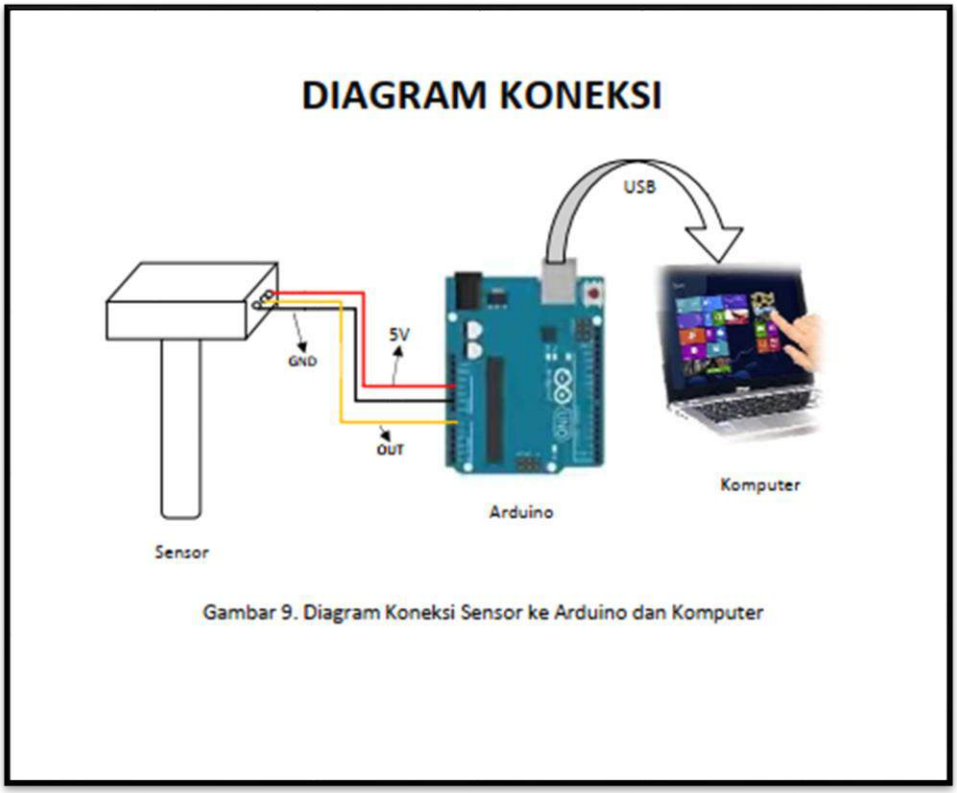
Grafik 6. Karakteristik nilai ADC sensor terhadap konsentrasi garam (uji TDS)



Grafik 7. Karakteristik TDS sensor kalibrator TDS meter terhadap konsentrasi garam (uji TDS)



Grafik 8. Karakteristik nilai ADC sensor terhadap nilai TDS sensor kalibrator TDS meter (uji TDS)



Gambar 9. Diagram Koneksi Sensor ke Arduino dan Komputer

A. DATA UJI KONDUKTIVITAS

N0	Garam Yodium(gr)	Air (ml)	ADC Sensor	Konduktivitas (µS)
1	4	100	431	588
2	8	100	446	589
3	12	100	453	593
4	16	100	466	594
5	20	100	468	595
6	24	100	478	597
7	28	100	482	598
8	32	100	485	600
9	36	100	496	601

Tabel 10. Data uji konduktivitas

Didapatkan persamaan :

y = 0.2142x + 494.93, dimana : x = nilai ADC, dan y=konduktivitas

B. DATA UJI TDS

N0	Garam Non Yodium (gr)	Air (ml)	ADC Sensor	TDS (ppm)
1	50	100	456	436
2	100	100	498	449
3	150	100	518	458
4	200	100	537	469
5	250	100	561	474
6	300	100	621	491

Tabel 11. Data uji TDS

Didapatkan persamaan :

y= 0.3417x + 281.08, dimana : x = nilai ADC, dan y=TDS

LAMPIRAN GAMBAR

A. Uji Konduktivitas (sampel menggunakan garam yodium)

Gambar 12. Bahan uji garam yodium yang dilarutkan

Gambar 13. Percobaan larutan garam yodium menggunakan sensor uji

Gambar 14. Sensor kalibrator ConduCtivity Meter

Gambar 15. Percobaan larutan garam yodium menggunakan sensor kalibrator Conductivity Meter

B. Percobaan TDS (sampel menggunakan garam non yodium)

Garam non yodium dipilih untuk mendapatkan nilai TDS yang signifikan

Gambar 16. Bahan uji garam non yodium yang dilarutkan

Gambar 17. Percobaan larutan garam non yodium menggunakan sensor uji

Gambar 18. Sensor kalibrator TDS Meter

Gambar 19. Percobaan larutan garam non yodium menggunakan sensor kalibrator TDS Meter

2. Sensor pH

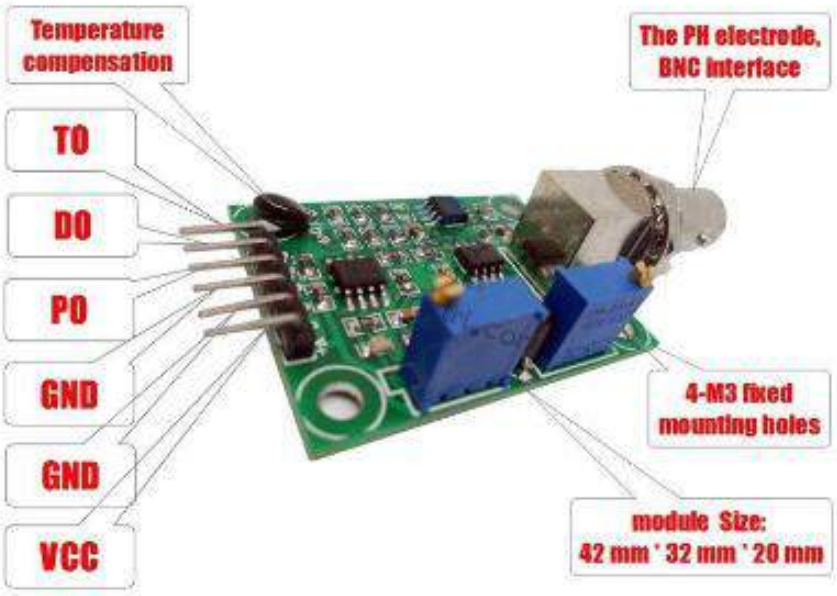
How to use a PH probe and sensor

If you worked with PH metering before you will know that PH values range from 0-14. Where PH 0 Will be very acidic, PH 7 will be neutral and PH 14 very alkaline. Water is near a PH 7 and this is usually around here that we will need to monitor PH of many things. A swimming pool, for example, should be slightly alkaline at 7.2, hydroponics systems around 6 (for optimum plant nutrition takeup) and aquaponics around 6.8.

I wrote this **PH probe and sensor** "how to" because it is not as straightforward as one would think (but quite easy when you understand the ins and outs) mostly because there is not a lot of information on this on the Internet, surely not detailed information.

We will first look at the ph probe module board and then the PH probe because both the PH probe and sensor have to be set correctly:

- offset setting
- limit setting
- sketch to test the board analogue range
- sketch for PH reading and calibration.
- calibration of PH probe
- PH probe usage
  The ph probe module in this tutorial is available on our site here: PH probe module BNC conector



PH Probe Sensor Pinout

TO – Temperature output
DO – 3.3V Output (from ph limit pot)
PO – PH analog output ==> **Arduino A0**
Gnd – Gnd for PH probe (can come from Arduino GND pin) ==> **Arduino GND**
Gnd – Gnd for board (can also come from Arduino GND pin) ==> **Arduino GND**
VCC – 5V DC (can come from Arduino 5V pin) ==> **Arduino 5V pin**
POT 1 – Analog reading offset (Nearest to BNC connector)
POT 2 – PH limit setting

PH probe module Offset and how to use it.

This board have the ability to supply a voltage output to the analogue board that will represent a PH value just like any other sensor that will connect to an analog pin. Ideally, we want a PH 0 represent 0v and a PH of 14 to represent 5V.

BUT there is a catch……, this board by default have PH 7 set to 0V (or near it, it differs from one PH probe to another, that is why we have to calibrate the probe as you will see later on), This means that the voltage will go into the minuses when reading acidic PH values and that cannot be read by the analog Arduino port. The offset pot is used to change this so that a PH 7 will read the expected 2.5V to the Arduino analog pin, the analog pin can read voltages between 0V and 5V hence the 2.5V that is halfway between 0V and 5V as a PH 7 is halfway between PH 0 and PH 14,

You will need to turn the offset potentiometer to get the right offset, The offset pot is the blue pot nearest to the BNC connector.

To set the offset is easy. First, you need to disconnect the probe from the circuit and short-circuit the inside of the BNC connector with the outside to simulate a neutral PH (PH7). I took a piece of wire, strip both sides, wrap the one side around the outside of the BNC connector and push the other side into the BNC hole. This short-circuit represents about a neutral PH reading of 7.

There are two ways you can do the adjustment.

If you have a multimeter handy you can measure the value of the PO pin and adjust the offset potentiometer until PO measures 2.5V.

I prefer to just use the sketch below. Just download it to your Arduino as you will with any other sketch, open serial monitor and view the reading there. All this sketch does is to print the volts it receives from the analog pin and print it to the serial monitor. It of course first changes the digital value to volts to make it easier. Now simply turn the offset pot until it is exactly 2.5V. You can learn more about reading voltages and digital representation of volts here: https://www.arduino.cc/en/Tutorial/ReadAnalogVoltage

Offset sketch

```
void setup() {
 // initialize serial communication at 9600 bits per second:
 Serial.begin(9600);
}

// the loop routine runs over and over showing the voltage on A0
void loop() {
 // read the input on analog pin 0:
 int sensorValue = analogRead(A0);
 // Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):
 float voltage = sensorValue * (5.0 / 1023.0);
 // print out the value you read:
 Serial.println(voltage);
 delay(300);
}
```

3.sensor ultrasonic waterroof

Sensor Ultrasonik Tahan Air JSN-SR04T

Selain modul pengukur jarak ultrasonik HC-SR04 yang sudah dikenal, modul lain yang terlihat di pasar daring adalah JSN-SR04 (dan variannya). Modul pengukur jarak ultrasonik JSN-SR04 memiliki beberapa perbedaan penting – modul ini memiliki satu probe transduser ultrasonik kedap air dan sirkuit elektronik khusus untuk menangani transduser. Selain itu, transduser dilengkapi dengan kabel tahan cuaca yang cukup panjang (~2,5 meter).

Menurut banyak blogpost, pengukur jarak ultrasonik JSN-SR0T4 dapat menyediakan fungsi penginderaan jarak nonkontak 25cm-450cm, dengan akurasi jangkauan hingga 2mm. Perlu dicatat bahwa hanya probe transduser dan bagian luarnya yang kedap air. Jika air masuk ke modul elektronik, perangkat dapat berhenti bekerja atau rusak secara permanen. ⚠

Saya pikir untuk beberapa proyek otomotif/robotik, sensor JSN-SR04 lebih cocok dibandingkan dengan sensor HC-SR04 biasa. Harganya, meskipun lebih mahal, tidak masalah jika Anda mempertimbangkan fitur unik yang Anda dapatkan!

Tegangan kerja JSN-SR04 adalah DC 5V dan arus kerja statis (arus diam) sekitar 5mA sedangkan arus kerja (arus operasi) sekitar 30mA. Frekuensi emisi akustik adalah 40kHz, dan sudut pengukuran adalah 45-75 derajat.

Modul (sensor/elektronik kontrol) memiliki empat koneksi yang sama dengan modul HR-SR04:

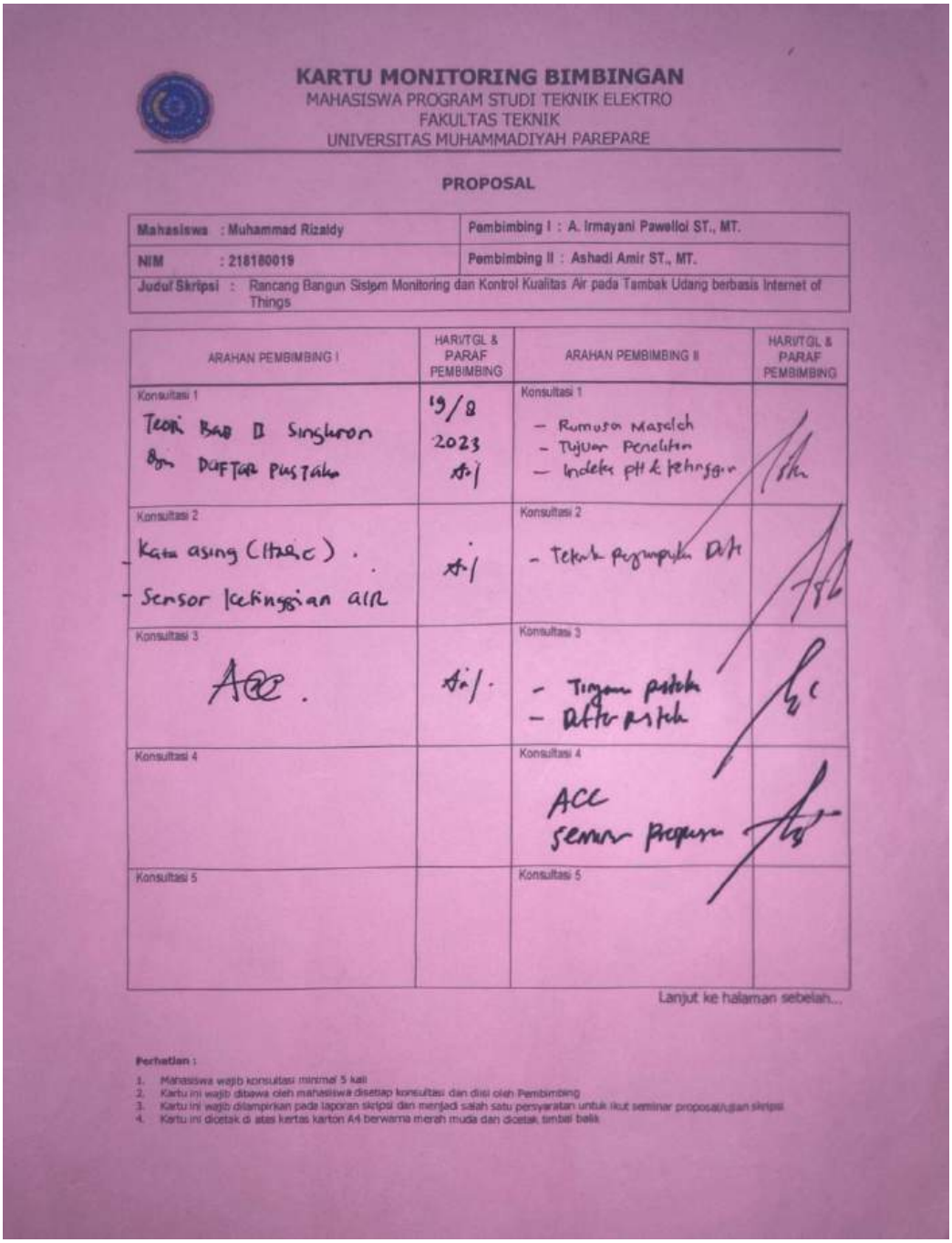
Pemicu, Gema, 5V dan Ground.

Komponen Utama & Lembar Data:

- STM8s003F3P6 https://www.st.com/resource/en/datasheet/stm8s003f3.pdf
- LMV324I https://www.st.com/resource/en/datasheet/lmv324.pdf
- J3Y (S8050) https://www.openimpulse.com/blog/wp-content/uploads/wpsc/downloadables/S8050-NPN-Transistor-SOT-23-Datasheet.pdf
- 2TY (S8550) https://www.rscomponents.kiev.ua/datasheets/s8550-2ty-datasheet.pdf

Lampiran- 4 Kartu monitoring bimbingan

1. Kartu Monitoring Bimbingan Proposal



KARTU MONITORING BIMBINGAN
MAHASISWA PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

PROPOSAL

Mahasiswa : Muhammad Rizaldy	Pembimbing I : A. Irmayani Pawelloi ST., MT.
NIM : 218180019	Pembimbing II : Ashadi Amir ST., MT.
Judul Skripsi : Rancang Bangun Sistem Monitoring dan Kontrol Kualitas Air pada Tambak Udang berbasis Internet of Things	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 Teori Bab II Singkron dgn Daftar Pustaka	19/8 2023	Konsultasi 1 - Rumusan Masalah - Tujuan Penelitian - Indeks ptt & tahapan	
Konsultasi 2 Kata asing (Italic). - Sensor ketinggian air		Konsultasi 2 - Teknik pengumpulan Data	
Konsultasi 3 ACC.		Konsultasi 3 - Tinjauan pustaka - Daftar pustaka	
Konsultasi 4		Konsultasi 4 ACC seminar proposal	
Konsultasi 5		Konsultasi 5	

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 A4 berwarna merah muda dan dicetak bintal 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, ...

Mengetahui
Ketua Program Studi

Asrul, ST., MT.
NBM. 986 836

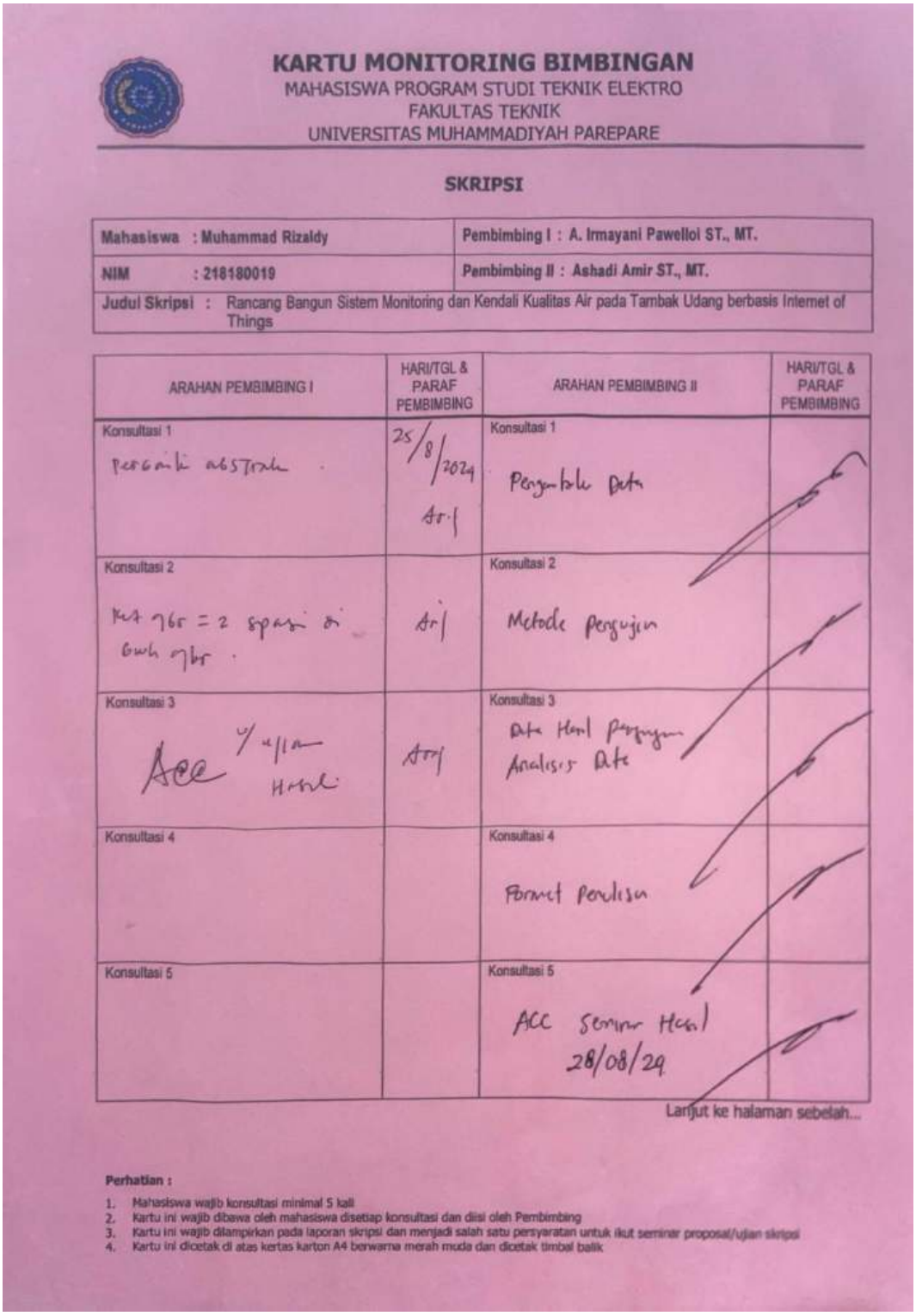
Mahasiswa

Muhammad Rizaldy
NIM. 218180019

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 A4 berwarna merah muda dan dicetak timbal balik

2. Kartu Monitoring Bimbingan Skripsi



KARTU MONITORING BIMBINGAN
MAHASISWA PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

SKRIPSI

Mahasiswa : Muhammad Rizaldy	**Pembimbing I** : A. Irmayani Pawelloi ST., MT.
NIM : 218180019	**Pembimbing II** : Ashadi Amir ST., MT.
Judul Skripsi : Rancang Bangun Sistem Monitoring dan Kendali Kualitas Air pada Tambak Udang berbasis Internet of Things	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 perbaiki abstrak	25/8/2024 Ar	Konsultasi 1 Pengolahan Data	
Konsultasi 2 Ket gbr = 2 spasi di bwh gbr	Ar	Konsultasi 2 Metode pengujian	
Konsultasi 3 Acc Ujian Hasil	Ar	Konsultasi 3 Data Hasil pengujian Analisis Data	
Konsultasi 4		Konsultasi 4 Format penulisan	
Konsultasi 5		Konsultasi 5 Acc Seminar Hasil 28/08/24	

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 A4 berwarna merah 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, 30 / 8 / 2024

Mengetahui
Ketua Program Studi

Asrul, ST., MT.
NBM. 986 836

Mahasiswa

Muhammad Rizaldy
NIM. 218180019

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 A4 berwarna merah muda dan dicetak timbal balik