

DAFTAR PUSTAKA

- [1] Z. I. Jumriah Langkong, Nandi K Sukendar, “Studi Pembuatan Minuman Isotonik Berbahan Baku Air Kelapa Tua (*Cocos Nicifera* L) Dan Ekstrak Belimbing Wuluh (*Avverhoa Bilimbi* L) Menggunakan Metode Sterilisasi Non-Thermalselama Penyimpanan,” *Canrea J. Food Technol. Nutr. Culin. J.*, pp. 53–62, 2018.
- [2] Z. H. Nasution, I. Suhaidi, and L. N. Limbong, “Pengaruh Perbandingan Air Kelapa Tua Dengan Sari Sirsak Dan,” vol. 4, no. 4, pp. 517–524, 2016.
- [3] C. Maria Dimova and P. M. R. Stirk, “Pengaruh Penambahan Ekstrak Etanol Dari Gambir Terhadap Sifat Kimia Air Kelapa Selama Penyimpanan Suhu Kamar,” pp. 9–25, 2019.
- [4] E. Mufida, R. S. Anwar, R. A. Khodir, and I. P. Rosmawati, “Perancangan Alat Pengontrol pH Air Untuk Tanaman Hidroponik Berbasis Arduino Uno,” *INSANtek*, vol. 1, no. 1, pp. 13–19, 2020, [Online]. Available: <http://ejournal.bsi.ac.id/ejurnal/index.php/insantek%0Ahttps://ejournal.bsi.ac.id/ejurnal/index.php/insantek>
- [5] F. Ariska, I. Hadi, and L. Lindawati, “Rancang Bangun Alat Pendeteksi Kelayakan Air Menggunakan Sensor pH,” *Jurasik (Jurnal Ris. Sist. Inf. dan Tek. Inform.*, vol. 4, no. 1, p. 127, 2019, doi: 10.30645/jurasik.v4i1.125.
- [6] Y. Rahmanto, A. Rifaini, S. Samsugi, and S. D. Riskiono, “Sistem Monitoring pH Air Pada Aquaponik Menggunakan Mikrokontroler Arduino Uno,” *J. Teknol. dan Sist. Tertanam*, vol. 1, no. 1, p. 23, 2020, doi: 10.33365/jtst.v1i1.711.
- [7] H. Amri, “Sistem Pengukuran Kualitas Air Bersih Berbasis Mikrokontroler Arduino,” *Pros. Semin. Nas. Fis. Univ. Riau ke-3*, no. September, pp. 27–30, 2018.
- [8] A. M. Muslimin, K. Karim, and T. W. Tria Winarsih, “Perancangan Alat Ukur pH Digital Berbasis Mikrokontroler Atmega32u4 Menggunakan Sensor Ph Meter V1.1,” *J. Nat.*, vol. 18, no. 1, pp. 36–44, 2022, doi: 10.30862/jn.v18i1.175.

- [9] M. Ariyanti, C. Suherman, Y. Maxiselly, and S. Rosniawaty, "Pertumbuhan Tanaman Kelapa (*Cocos Nucifera L.*) Dengan Pemberian Air Kelapa," *J. Hutan Pulau-Pulau Kecil*, vol. 2, no. 2, pp. 201–212, 2018, doi: 10.30598/jhppk.2018.2.2.201.
- [10] P. K. S. Farah Meita Pratiwi, "Etnobotani Kelapa (*Cocos Nucifera L.*) Di Wilayah Denpasar Dan Badung (Etnobotany)," *J. SIMBIOSIS I*, vol. 2, no. September, pp. 1–23, 2016.
- [11] P. Eastern and C. Utama, "Sumber: PT. Pasific Eastern Coconut Utama 4," no. 0265, pp. 4–15, 2010.
- [12] L. N. Ikhsanto, "Sistem Pengecekan pH Air Otomatis Menggunakan Sensor pH Probe Berbasis Arduino Pada Sumur Bor," vol. 21, no. 1, pp. 1–9, 2020.
- [13] Y. B. Wariyanti, "Kualitas Air Kelapa Hijau 9cocos Nucifera L) Berdasarkan Perbedaan Umur Buah Kelapa." pp. 75–82, 2008.
- [14] P. Y. P. Pratama, "Perancangan PH Meter Dengan Sensor PH Air Berbasis Arduino," vol. 3, no. 2, 2022.
- [15] DFRobot, "Datasheet PH meter (SKU : SEN0161)," *Wiki DFRobot*, pp. 1–20, 2021, [Online]. Available: https://wiki.dfrobot.com/PH_meter_SKU__SEN0161_
- [16] A. Appiani, "Datasheet Arduino MEGA 2560," *Arduino ® MEGA 2560*, pp. 1–18, 2022, [Online]. Available: www.arduino.cc
- [17] Barmawi, Azhar, and M. Kamal, "Rancang Bangun Penggunaan Sensor Thermocouple Pada Alat Penyangrai Biji Coklat Secara Otomatis Berbasis Arduino Uno," *J. Tektro*, vol. 7, no. 1, p. 2023, 2023.
- [18] "Datasheet Motor DC RS550 63," pp. 63–69, 2019.
- [19] M. M. Wurfel, "Datasheet SG996R High Torque Metal Gear Dual Ball Bearing Servo," *Electonic Caldas*, no. 6 V, pp. 1–2, 2015.
- [20] A. Hidayah, V. A. Bahri, Isminarti, and M. Nur, "Rancang Bangun Alat Pemilah Sampah dan Pengolahan Limbah Kaleng Minuman Berbasis Mikrokontroler," *Mechatronics J. Prof. Entrep.*, vol. 1, pp. 7–12, 2019, [Online]. Available: <http://jurnal.politeknikbosowa.ac.id/index.php/JMAPLE/article/view/256%0Ahttp://jurnal.politeknikbosowa.ac.id/index.php/JMAPLE/article/download/256/110>

- [21] R. O. G. Vwduw and R. O. G. V Wduw, "Datasheet Power Supply," p. 8.
- [22] Rajaguru Electronics, "Datasheet R385 Diaphragm Mini Water Pump 12VDC," pp. 7–8, 2017, [Online]. Available: <https://5.imimg.com/data5/SELLER/Doc/2021/4/RH/JK/XD/1833510/r385-dc-9-12v-water-pump-maximum-lift-3-meters.pdf>
- [23] S. Afra, N. Wahyudin, R. Vahlevi, H. Prayoga, and N. Prasetyo, "Sistem Kontrol Pemanas Air Menggunakan Sensor Ultrasonik Dan Arduino Uno," *Power Elektron. J. Orang Elektro*, vol. 9, no. 2, pp. 30–35, 2020, doi: 10.30591/polektr.v9i2.2017.
- [24] H. A. Irmayani, Asrul, and M. N. Kaliky, "Desain Bangun Ayakan Alat Mesin Tanaman Perkebunan," *J. Telekomun. Kendali dan List.*, vol. 2, no. 1, pp. 12–22, 2020.
- [25] H. Technology, "User Guide - 4 Channel 5V Optical Isolated Relay Module," *Occup. Heal. Saf.*, vol. 74, no. 2, p. 24, 2015, [Online]. Available: <http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=16274161&site=ehost-live>
- [26] A. Laksana, A. Amir, T. S. Sollu, and M. A. Indrajaya, "Perancangan Sistem Antrian Anti Covid 19 Berbasis Internet Of Things Menggunakan Raspberry Pi W Zero," *Foristek*, vol. 13, no. 2, pp. 58–66, 2022, doi: 10.54757/fs.v13i2.148.
- [27] "Datasheet Buzzer," p. 16, 2012.
- [28] I. Boimau, R. Irmawanto, and M. F. Taneo, "Rancang Bangun Alat Ukur Laju Bunyi Di Udara Menggunakan Sensor Ultrasonik Berbasis Arduino," *Cyclotron*, vol. 2, no. 2, 2019, doi: 10.30651/cl.v2i2.3253.
- [29] E. J. Morgan, "HY-SRF05 Precision Ultrasonic Sensor," *Datasheetpdf*, pp. 4–6, 2016, [Online]. Available: <http://forum.arduino.cc/index.php?topic=106043.0>

LAMPIRAN A

Program Arduino

```
#define pin_limit_a 22
#define pin_limit_b 23
#define pin_limit_drill 26
#define pin_motor_rotate_a 8
#define pin_motor_rotate_b 9
#define pin_motor_drill_a 10
#define pin_motor_drill_b 11
#define pin_servo_grip_a 4
#define pin_servo_grip_b 5
#define pin_servo_feeder_a 6
#define pin_servo_feeder_b 7
#define pin_relay_drill 50
#define pin_relay_pump_a 51
#define pin_relay_pump_b 52
#define pin_ultrasonic_trigger 28
#define pin_ultrasonic_echo 29
#define pin_buzzer 53

#define servo_grip_a_pos_start 8
#define servo_grip_a_pos_end 25
#define servo_grip_b_pos_start 175
#define servo_grip_b_pos_end 138
#define servo_feeder_a_pos_close 130
#define servo_feeder_a_pos_open 45
#define servo_feeder_b_pos_close 140
#define servo_feeder_b_pos_open 45

#define rotary_pos_middle 25
#define rotary_pos_end 42

void setup() {
  Serial.begin(9600);
  setupLimit();
  setupServo();
  setupMotor();
  setupRelay();
```

```

    setupBuzzer();
    setupLcd();
}

float lastPh = 0;
void loop() {
    lcdPrint(0, "pH terakhir :");
    lcdPrint(1, String(lastPh));
    while (ultrasonicGet() < 5) {
        buzzerOn();
        delay(1000);
        buzzerOff();
        delay(1000);
    }
    buzzerOff();
    motorDrillPull();
    while (!isLimitDrill()) {
        delay(100);
        serialEvent();
    }
    motorDrillStop();
    motorRotateUp();
    while (!isLimitB()) {
        delay(100);
        serialEvent();
    }
    motorRotateStop();
    encoderReset();
    while (!isLimitA()) {
        delay(100);
        serialEvent();
    }
    servoFeed();
    delay(500);
    servoGripOn();
    delay(500);
    motorRotateDown();
    while (encoderGetValue() < rotary_pos_middle) {}
    motorRotateStop();
}

```

```

drilling();
motorRotateDown();
while (encoderGetValue() < rotary_pos_end) {}
motorRotateStop();
servoGripOff();
sorting();
}

```

```

void drilling() {
  relayDrillOn();
  motorDrillPush();
  delay(33000);
  motorDrillPull();
  while (!isLimitDrill()) {
    delay(100);
    serialEvent();
  }
  relayDrillOff();
  motorDrillStop();
}

```

```

void sorting() {
  int stableCount = 0;
  float lastValue = 0;
  float value = 0;
  while (stableCount < 2) {
    value = phGet();
    if (String(value) == String(lastValue)) {
      stableCount++;
    }
    Serial.print(value);
    Serial.print("\t");
    Serial.print(lastValue);
    Serial.print("\t");
    Serial.print(stableCount);
    Serial.print("\n");
    lastValue = value;
    lcdPrint(0, "pH :");
    lcdPrint(1, String(value));
  }
}

```

```

    delay(100);
}
stableCount = 0;
if (value > 5.2) {
    relayPumpAOn();
    while (isWater1() == false) {
        relayPumpAOn();
        delay(100);
    }
    delay(1000);
    while (stableCount < 10) {
        if (isWater1() == false) {
            stableCount++;
        }
        relayPumpAOn();
        Serial.print(stableCount);
        Serial.print('\n');
        delay(100);
    }
} else {
    relayPumpBOn();
    while (isWater2() == false) {
        relayPumpBOn();
        delay(100);
    }
    delay(1000);
    while (stableCount < 10) {
        if (isWater2() == false) {
            stableCount++;
        }
        relayPumpBOn();
        Serial.print(stableCount);
        Serial.print('\n');
        delay(100);
    }
}
relayPumpAOff();
relayPumpBOff();
lastPh = value;

```

```

}

void serialEvent() {
  if (Serial.available() > 0) {
    String data = Serial.readStringUntil('\n');
    Serial.println(data);
    if (data.indexOf("w") != -1) {
      isWater1();
    } else if (data.indexOf("f") != -1) {
      servoFeed();
    } else if (data.indexOf("s") != -1) {
      sorting();
    } else if (data.indexOf("ph") != -1) {
      Serial.println(phGet());
    } else if (data.indexOf("p1") != -1) {
      relayPumpAOn();
    } else if (data.indexOf("p2") != -1) {
      relayPumpBOn();
    } else if (data.indexOf("0") != -1) {
      relayPumpAOff();
      relayPumpBOff();
    }
  }
}
}
}

```


BUZZER

```
void setupBuzzer() {  
  pinMode(pin_buzzer, OUTPUT);  
}
```

```
void buzzerOn() {  
  digitalWrite(pin_buzzer, HIGH);  
}
```

```
void buzzerOff() {  
  digitalWrite(pin_buzzer, LOW);  
}
```

LCD

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
void setupLcd() {
  lcd.init();
  lcd.backlight();
  lcd.clear();
}
```

```
void lcdPrint(int line, String msg) {
  lcd.setCursor(0, line);
  lcd.print(" ");
  lcd.setCursor(0, line);
  lcd.print(msg);
}
```

LIMIT SWITCH

```
void setupLimit() {  
  pinMode(pin_limit_a, INPUT_PULLUP);  
  pinMode(pin_limit_b, INPUT_PULLUP);  
  pinMode(pin_limit_drill, INPUT_PULLUP);  
}
```

```
boolean isLimitA() {  
  return !digitalRead(pin_limit_a);  
}
```

```
boolean isLimitB() {  
  return !digitalRead(pin_limit_b);  
}
```

```
boolean isLimitDrill() {  
  return !digitalRead(pin_limit_drill);  
}
```

MOTOR

```
void motorRotateDown() {  
  analogWrite(pin_motor_rotate_a, 200);  
  analogWrite(pin_motor_rotate_b, 0);  
}
```

```
void motorRotateStop() {  
  analogWrite(pin_motor_rotate_a, 0);  
  analogWrite(pin_motor_rotate_b, 0);  
}
```

```
void motorDrillPush() {  
  analogWrite(pin_motor_drill_a, 0);  
  analogWrite(pin_motor_drill_b, 255);  
}
```

```
void motorDrillPull() {  
  analogWrite(pin_motor_drill_a, 255);  
  analogWrite(pin_motor_drill_b, 0);  
}
```

```
void motorDrillStop() {  
  analogWrite(pin_motor_drill_a, 0);  
  analogWrite(pin_motor_drill_b, 0);  
}
```

SENSOR PH

```
long samples = 100;
```

```
float phGet() {  
  int measurements = 0;  
  for (int i = 0; i < samples; i++) {  
    measurements += analogRead(A0);  
    delay(10);  
  }  
  float measuring = measurements / samples;  
  // kalibrasi 1  
  float voltage = 5 / 1024.0 * measuring;  
  return (7.97 + (0.986 * voltage) + (-3.98 * pow(voltage, 2))) + 1;  
  // // kalibrasi 2  
  // return -0.0266 * measuring + 12.4;  
}
```

RELAY

```
void setupRelay() {  
  pinMode(pin_relay_drill, OUTPUT);  
  pinMode(pin_relay_pump_a, OUTPUT);  
  pinMode(pin_relay_pump_b, OUTPUT);  
  digitalWrite(pin_relay_drill, HIGH);  
  digitalWrite(pin_relay_pump_a, HIGH);  
  digitalWrite(pin_relay_pump_b, HIGH);  
}
```

```
void relayDrillOn() {  
  digitalWrite(pin_relay_drill, LOW);  
}
```

```
void relayDrillOff() {  
  digitalWrite(pin_relay_drill, HIGH);  
}
```

```
void relayPumpAOn() {  
  digitalWrite(pin_relay_pump_a, LOW);  
}
```

```
void relayPumpAOff() {  
  digitalWrite(pin_relay_pump_a, HIGH);  
}
```

```
void relayPumpBOn() {  
  digitalWrite(pin_relay_pump_b, LOW);  
}
```

```
void relayPumpBOff() {  
  digitalWrite(pin_relay_pump_b, HIGH);  
}
```

ROTARY ENCODER

```
#include <Encoder.h>

Encoder myEnc(24, 25);

long oldPosition = -999;

long encoderGetValue() {
    return myEnc.read();
}

void encoderReset() {
    myEnc.write(0);
}
```

SERVO

```
#include <Servo.h>

Servo servoGripA;
Servo servoGripB;
Servo servoFeederA;
Servo servoFeederB;

void setupServo() {
  servoGripA.attach(pin_servo_grip_a);
  servoGripB.attach(pin_servo_grip_b);
  servoFeederA.attach(pin_servo_feeder_a);
  servoFeederB.attach(pin_servo_feeder_b);
  servoGripA.write(servo_grip_a_pos_start);
  servoGripB.write(servo_grip_b_pos_start);
  servoFeederA.write(servo_feeder_a_pos_open);
  servoFeederB.write(servo_feeder_b_pos_close);
}

void servoFeed() {
  servoFeederA.write(servo_feeder_a_pos_close);
  delay(1000);

  servoFeederB.write(servo_feeder_b_pos_open);
  delay(1000);

  servoFeederB.write(servo_feeder_b_pos_close);
  delay(1000);
  servoFeederA.write(servo_feeder_a_pos_open);
}

void servoGripOn() {
  servoGripA.write(servo_grip_a_pos_end);
  servoGripB.write(servo_grip_b_pos_end);
}

void servoGripOff() {
  servoGripA.write(servo_grip_a_pos_start);
  servoGripB.write(servo_grip_b_pos_start);
}
```


ULTRASONIK

```
#include <Ultrasonic.h>

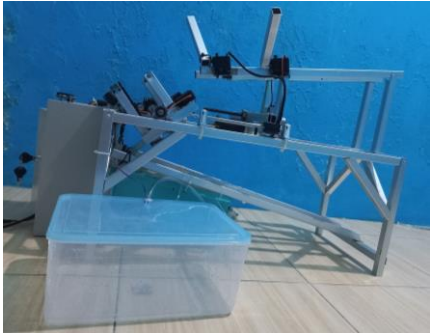


Ultrasonic ultrasonicIn(pin_ultrasonic_trigger, pin_ultrasonic_echo);


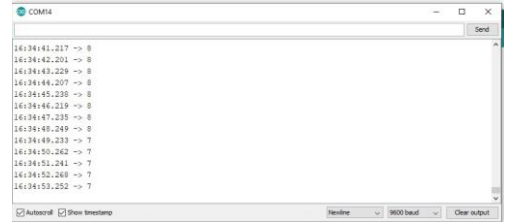
int ultrasonicGet() {
    return ultrasonicIn.read();
}

# WATER
bool isWater1() {
    if (analogRead(1) > 50) {
        return true;
    }
    return false;
}

bool isWater2() {
    if (analogRead(2) > 50) {
        return true;
    }
    return false;
}
```

LAMPIRAN B
Dokumentasi Pengujian

	<p>Hasil Perancangan Mekanik</p>
	<p>Hasil Penyortiran pH Air Kelapa $\geq 5,2$</p>
	<p>Hasil Penyortiran pH Air Kelapa $< 5,2$</p>

	<p>Tampilan LCD</p>
	<p>Hasil Pengukuran Menggunakan pH Meter Standar</p>
 <pre>COM4 16:34:41.217 -> 0 16:34:42.201 -> 0 16:34:43.229 -> 0 16:34:44.207 -> 0 16:34:45.230 -> 0 16:34:46.219 -> 0 16:34:47.235 -> 0 16:34:48.249 -> 0 16:34:49.233 -> 7 16:34:50.262 -> 7 16:34:51.241 -> 7 16:34:52.269 -> 7 16:34:53.252 -> 7 Autoscroll [x] Show timestamp Hexline 9600 baud Clear output</pre>	<p>Hasil Pengujian Ultrasonik Pada Serial Monitor</p>



Hasil Pengujian
Ultrasonik
Menggunakan Alat
Ukur Standar

BIODATA PENULIS



Nama : Riva Oktaviana
Tempat/Tanggal Lahir : Banyumas, 20 Oktober 2002
Alamat : Ds. Cikidang RT 01/RW 02, Kecamatan Cilogok, Kabupaten Banyumas, Jawa Tengah, 53162
Telepon/Hp : 0857-1317-2401
Hobi : Nonton Film
Motto : Hidup bukan saling mendahului, bermimpilah sendiri-sendiri.

Riwayat Pendidikan :

- SD Negeri 02 Cikidang Tahun 2008-2014
- SMP Negeri 1 Cilogok Tahun 2014-2017
- SMK Ma'arif NU 1 Ajibarang Tahun 2017-2020
- Politeknik Negeri Cilacap Tahun 2020-2023

Penulis telah mengikuti sidang Tugas Akhir pada tanggal 10 Agustus 2023 sebagai salah satu persyaratan untuk memperoleh gelar Ahli Madya (A.Md).