

# EcoBin

Inovasi Tempat Sampah Pintar Berbasis IoT untuk Pemilahan Sampah Otomatis dan Ramah Lingkungan



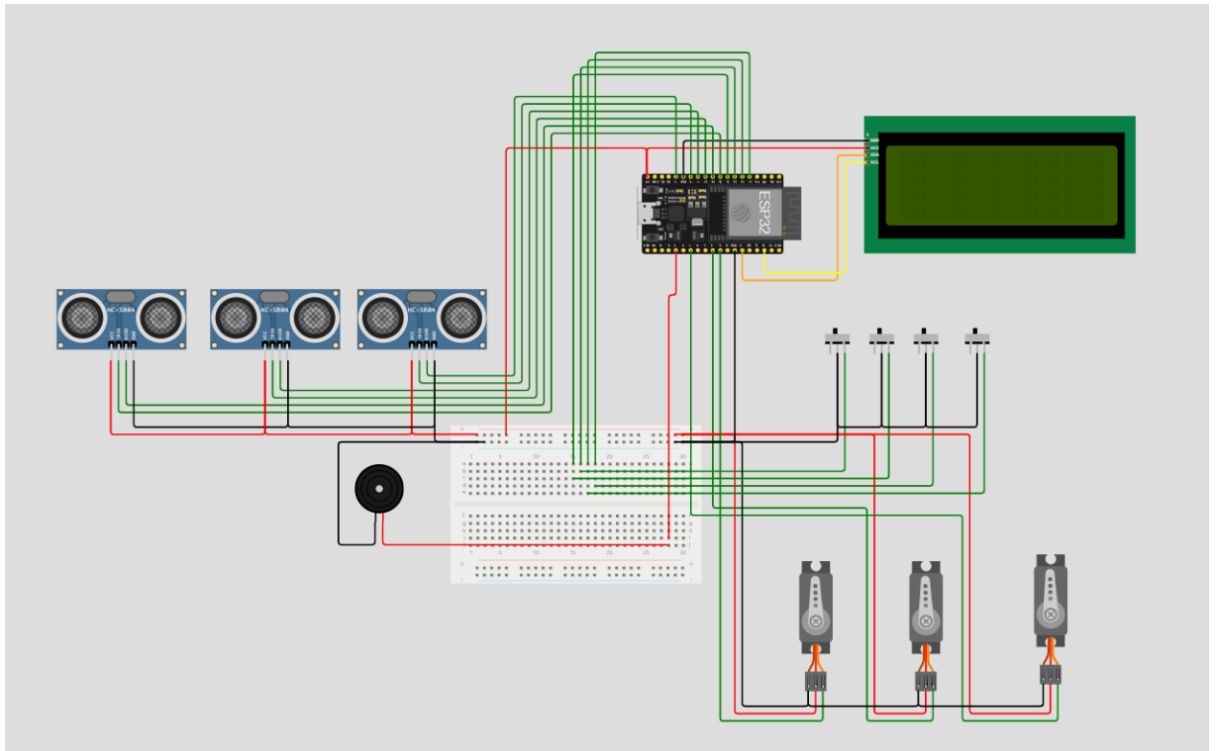
Nama Anggota Team:

- |  |                      |
|--|----------------------|
| 1. Regar Mukti Wijaya / 2122600032         | (Hardware Developer) |
| 2. Yunanta Adi Wijaya / 2122600034         | (Data Analyst)       |
| 3. Veda Alfa Elydal / 2122600039           | (Software Developer) |
| 4. Dwi Angga Ramadhani / 2122600053        | (UI/UX Designer)     |
| 5. Hanif Jauhar Islami / 2122600048        | (Project Manager)    |
| 6. Dewa Gede Angkasa Arinanta / 2122600059 | (3D Designer)        |

## Progress Laporan Mingguan

### Laporan Minggu Ke-9

- Desain Rangkaian menggunakan Wokwi  
Melakukan penyempurnaan desain pada aplikasi wokwi dan melakukan perbaikan pada error rangkaian yang masih terjadi. Berikut adalah tampilan dari rangkaian yang nantinya akan terpasang pada hardware komponen.



- Pemrograman rangkaian pada aplikasi Wokwi  
Detail pemrograman dapat dilihat seperti berikut, dengan hasil program yang nantinya akan ditampilkan pada README.md pada tampilan awal Github.

```
• #include <Wire.h>
• #include <LiquidCrystal_I2C.h>
• #include <ESP32Servo.h>
•
• // Inisialisasi alamat LCD I2C dan ukuran (20x4)
• LiquidCrystal_I2C lcd(0x27, 20, 4);
•
• // Definisi pin sensor ultrasonik
• const int trigPin1 = 25; // Pin Trigger untuk sampah organik
• const int echoPin1 = 26; // Pin Echo untuk sampah organik
• const int trigPin2 = 27; // Pin Trigger untuk sampah anorganik
• const int echoPin2 = 14; // Pin Echo untuk sampah anorganik
• const int trigPin3 = 12; // Pin Trigger untuk sampah metal
• const int echoPin3 = 13; // Pin Echo untuk sampah metal
•
```

```

• // Definisi Pin untuk push button dan servo
• const int pb1Pin = 32; // Push button 1 untuk servo1 (Metal)
• const int pb2Pin = 33; // Push button 2 untuk servo1
  (Anorganik/Organik)
• const int pb3Pin = 34; // Push button 3 untuk servo2 (Anorganik)
• const int pb4Pin = 35; // Push button 4 untuk servo2 (Organik)
• const int servoPin1 = 4; // Pin untuk servo1
• const int servoPin2 = 5; // Pin untuk servo2
• const int servoPin3 = 18; // Pin untuk servo3
•
• // Definisi pin untuk buzzer
• const int buzzerPin = 2;
•
• // Variabel kapasitas
• int kapasitasOrganik = 0;
• int kapasitasAnorganik = 0;
• int kapasitasMetal = 0;
•
• // Inisialisasi objek Servo
• Servo myServo1;
• Servo myServo2;
• Servo myServo3;
•
• // Fungsi untuk menginisialisasi LCD dan Serial
• void initializeDisplay() {
•   lcd.init();
•   lcd.backlight();
•   lcd.setCursor(0, 0);
•   lcd.print("Kapasitas Sampah");
•   lcd.setCursor(0, 1);
•   lcd.print("Organik   = ");
•   lcd.setCursor(0, 2);
•   lcd.print("Anorganik = ");
•   lcd.setCursor(0, 3);
•   lcd.print("Metal     = ");
• }
•
• // Fungsi untuk membaca jarak dari sensor ultrasonik
• int readDistance(int trigPin, int echoPin) {
•   digitalWrite(trigPin, LOW);
•   delayMicroseconds(2);
•   digitalWrite(trigPin, HIGH);
•   delayMicroseconds(10);
•   digitalWrite(trigPin, LOW);
•
•   long duration = pulseIn(echoPin, HIGH);
•   int distance = duration * 0.034 / 2;
•   return distance;

```

```

• }
•
• // Fungsi untuk menghitung kapasitas berdasarkan jarak
• int calculateCapacity(int distance) {
•     if (distance <= 10) return 100;
•     else if (distance <= 20) return 75;
•     else if (distance <= 30) return 50;
•     else if (distance <= 40) return 25;
•     else return 0;
• }
•
• // Fungsi untuk menampilkan kapasitas pada LCD
• void displayCapacity(int kapasitas, int row) {
•     lcd.setCursor(12, row);
•     lcd.print(" "); // Hapus nilai sebelumnya
•     lcd.setCursor(12, row);
•     lcd.print(kapasitas);
•     lcd.print("%");
•
•     if (kapasitas == 100) {
•         lcd.setCursor(16, row);
•         lcd.print("FULL");
•     } else {
•         lcd.setCursor(16, row);
•         lcd.print(" ");
•     }
• }
•
• // Fungsi untuk mengaktifkan buzzer dengan tone
• void activateBuzzer() {
•     tone(buzzerPin, 200);
•     delay(500);
•     noTone(buzzerPin);
• }
•
• void setup() {
•     // Inisialisasi LCD dan Serial
•     initializeDisplay();
•     Serial.begin(9600);
•
•     // Inisialisasi Pin
•     pinMode(trigPin1, OUTPUT);
•     pinMode(echoPin1, INPUT);
•     pinMode(trigPin2, OUTPUT);
•     pinMode(echoPin2, INPUT);
•     pinMode(trigPin3, OUTPUT);
•     pinMode(echoPin3, INPUT);
•     pinMode(buzzerPin, OUTPUT);

```

```

•
• // Menggunakan INPUT_PULLUP untuk push button
• pinMode(pb1Pin, INPUT_PULLUP);
• pinMode(pb2Pin, INPUT_PULLUP);
• pinMode(pb3Pin, INPUT_PULLUP);
• pinMode(pb4Pin, INPUT_PULLUP);
•
• // Attach servo ke pin
• myServo1.attach(servoPin1);
• myServo2.attach(servoPin2);
• myServo3.attach(servoPin3);
• myServo1.write(90);
• myServo2.write(90);
• myServo3.write(90);
• }
•
• bool sampahpenuh = false; // Status awal dari kondisi penuh
•
• void loop() {
• // Membaca status push button
• bool pb1State = digitalRead(pb1Pin) == LOW;
• bool pb2State = digitalRead(pb2Pin) == LOW;
• bool pb3State = digitalRead(pb3Pin) == LOW;
• bool pb4State = digitalRead(pb4Pin) == LOW;
•
• // Kontrol Servo1
• if (pb1State) {
• myServo1.write(180);
• Serial.println("PB1 Ditekan: Servo1 ke kanan Mendeteksi sampah
Metal");
• delay(1000);
• myServo1.write(90);
• } else if (pb2State) {
• myServo1.write(0);
• Serial.println("PB2 Ditekan: Servo1 ke kiri Mendeteksi sampah
Organik/Anorganik");
• delay(1000);
• myServo1.write(90);
• }
•
• // Kontrol Servo2
• if (pb3State && !pb4State) {
• myServo2.write(180); // Servo2 bergerak ke kanan
• Serial.println("PB3 Ditekan: Servo2 ke kanan Mendeteksi sampah
Anorganik");
• delay(1000);
• myServo2.write(90); // Servo kembali ke posisi tengah
• } else if (pb3State && pb4State) {

```

```

• myServo2.write(0); // Servo2 bergerak ke kiri
• Serial.println("PB3 dan PB4 Ditekan: Servo2 ke kiri Mendeteksi sampah organik");
• delay(1000);
• myServo2.write(90); // Servo kembali ke posisi tengah
• }
•
• // Baca jarak dari setiap sensor
• int distanceOrganik = readDistance(trigPin1, echoPin1);
• int distanceAnorganik = readDistance(trigPin2, echoPin2);
• int distanceMetal = readDistance(trigPin3, echoPin3);
•
• // Hitung kapasitas berdasarkan jarak
• kapasitasOrganik = calculateCapacity(distanceOrganik);
• kapasitasAnorganik = calculateCapacity(distanceAnorganik);
• kapasitasMetal = calculateCapacity(distanceMetal);
•
• // Tampilkan hasil kapasitas pada LCD
• displayCapacity(kapasitasOrganik, 1);
• displayCapacity(kapasitasAnorganik, 2);
• displayCapacity(kapasitasMetal, 3);
•
• // Debugging: Tampilkan jarak pada Serial Monitor
• Serial.print("Jarak Organik: ");
• Serial.print(distanceOrganik);
• Serial.print(" cm, Anorganik: ");
• Serial.print(distanceAnorganik);
• Serial.print(" cm, Metal: ");
• Serial.println(distanceMetal);
•
• // Cek apakah salah satu kapasitas mencapai 100%
• bool kondisiSampahPenuh = (kapasitasOrganik == 100 ||
kapasitasAnorganik == 100 || kapasitasMetal == 100);
• if (kondisiSampahPenuh) {
•   if (!sampahpenuh) {
•     sampahpenuh = true;
•     myServo3.attach(servoPin3);
•     myServo3.write(180);
•     Serial.println("Sampah Penuh, Servo3 bergerak ke kiri");
•     activateBuzzer();
•     delay(2000);
•     myServo3.detach();
•   }
• } else if (sampahpenuh) {
•   myServo3.attach(servoPin3);
•   myServo3.write(90);
•   Serial.println("Kapasitas Tidak Penuh, Servo3 kembali ke posisi awal");

```

```

•   delay(500); // Beri waktu untuk servo mencapai posisi
•
•   // Koreksi posisi jika ada error
•   int currentAngle = myServo3.read(); // Membaca posisi aktual servo
•   if (currentAngle != 90) {
•       Serial.println("Koreksi posisi servo...");
•       myServo3.write(90); // Kembali ke posisi yang seharusnya (90
derajat)
•       delay(200); // Tambahkan sedikit delay untuk memastikan servo
selesai bergerak
•   }
•
•   myServo3.detach();
•   sampahpenuh = false;
•   }
• }

```

- Penyempurnaan tampilan README.md

Penambahan informasi mengenai progress update system pada EcoBin seperti penambahan daftar isi, informasi anggota, update skematik rangkaian, serta penambahan tampilan desain awal 3D untuk alat EcoBin.