**SISTEM OTOMATISASI PETERNAKAN IKAN**

**A logo with text and images

Description automatically generated**

Kelompok : 2

Anggota :

* Hafizh Nabhan Robbani – X-I
* Faqih Ahmad Prasetyo – XI-D
* Rakha Pratama Hermawan – X-H
* Zahra Tusyifa – X-I
* Gabriel Anggi Silalahi – X-F
* Tyas Chezza Azzahra – XI-J
* Fachri Irsyad Suherman – XI-F

**TAHAP 1 – STUDI LITERATUR**

1. **Spesifikasi**
2. Arduino Uno R3

Berfungsi sebagai mikrokontroler yang mengontrol keseluruhan proses, mulai dari pembacaan sensor hingga pengendalian aktuator.

1. Mini Servo

Servo digunakan untuk menggerakkan feeder agar dapat memberi pakan pada waktu yang telah ditentukan.

1. Ultrasonic HC-SR04

Sensor ini mengukur ketinggian air di kolam dengan mengirimkan gelombang ultrasonik dan menghitung waktu pantulan kembali untuk menentukan jaraknya.

1. LCD I2C (16x2)

LCD akan menampilkan jarak yang diukur oleh sensor ultrasonik dalam satuan cm. Komponen ini memberikan informasi visual untuk memudahkan pengguna dalam memantau sistem.

1. Kabel Jumper

Kabel jumper memungkinkan setiap komponen (seperti sensor, relay, servo, dan LCD) terhubung ke pin-pin Arduino untuk menjalankan fungsinya masing-masing.

1. Power supply

Mengalirkan arus listrik yang stabil ke semua komponen agar alat dapat berjalan sesuai fungsinya dan upload programnya

1. Bread Board

Menyalurkan arus listrik antar komponen melalui kabel jumper, Merancang sirkuit mikrokontroler seperti Arduino, Membangun purwarupa sirkuit elektronik semipermanen.

1. Resistor

mengontrol arus besar dan menurunkan tingkat tegangan dalam berbagai aplikasi elektronik . ccontohnya pada LED pada projek ini.

1. LED

Memberikan indikator visual terkait kondisi ketinggian air:

* **Merah**: Air terlalu rendah.
* **Kuning**: Air sedang.
* **Hijau**: Air cukup tinggi.

1. **Cara kerja**
2. **Pengukuran Ketinggian Air**:

* Sensor ultrasonik memancarkan gelombang ultrasonik dari pin **Trig**.
* Gelombang ini akan memantul kembali dari permukaan air dan diterima oleh pin **Echo**.
* Waktu tempuh gelombang dihitung untuk mendapatkan jarak antara sensor dan permukaan air.
* Nilai jarak ini diolah menjadi nilai ketinggian air (dalam cm).

1. **Indikasi Ketinggian Air:**

* Nilai ketinggian air ditampilkan di LCD secara real-time.
* Berdasarkan nilai ketinggian air:
  + **< 5 cm**: LED merah menyala, menandakan air rendah.
  + **5 – 8 cm**: LED kuning menyala, menandakan air dalam kondisi sedang.
  + **>= 9 cm**: LED hijau menyala, menandakan air cukup tinggi.

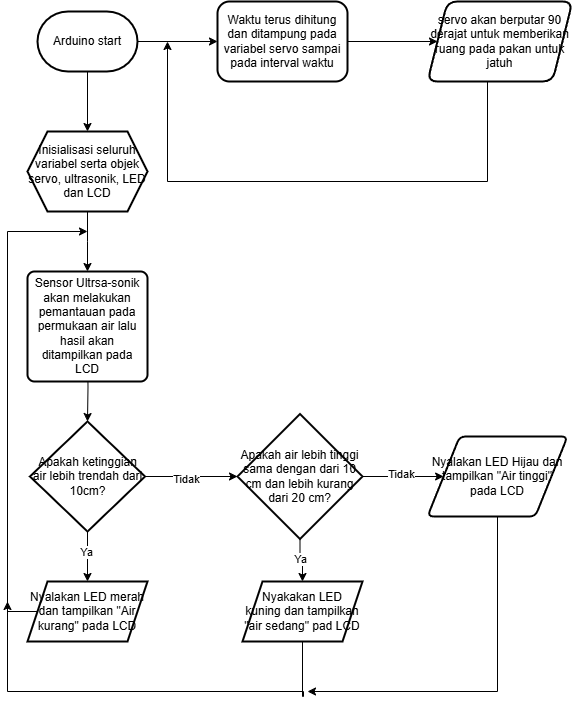
1. **Pemberian Pakan Otomatis:**

* Servo akan bergerak ke posisi **90°** (membuka pintu pakan) setiap **10 detik selama demo,** dan memungkinkan untuk diubah ke jadwal real untuk pakan ikan seperti 12 jam sekali.
* Setelah membuka selama **2 detik**, servo kembali ke posisi **0°** (tertutup).

1. **Pengendalian LED dan LCD:**

* Kondisi air divisualisasikan menggunakan LED dan ditampilkan secara rinci di LCD.
* LCD akan memberikan informasi kondisi air, seperti "Air Rendah", "Air Sedang", atau "Air Tinggi".

1. **Alur Program (Flowchart)**

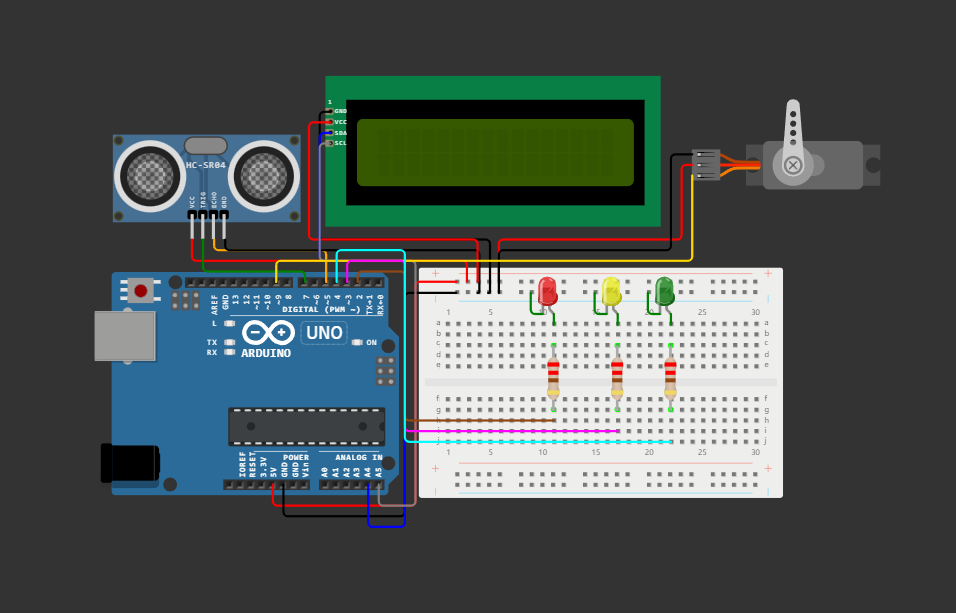


**TAHAP 2 – SIMULASI**

1. **Link simulasi**

(hhttps://wokwi.com/projects/420406378103456769)

1. **Rangkaian Sistem**

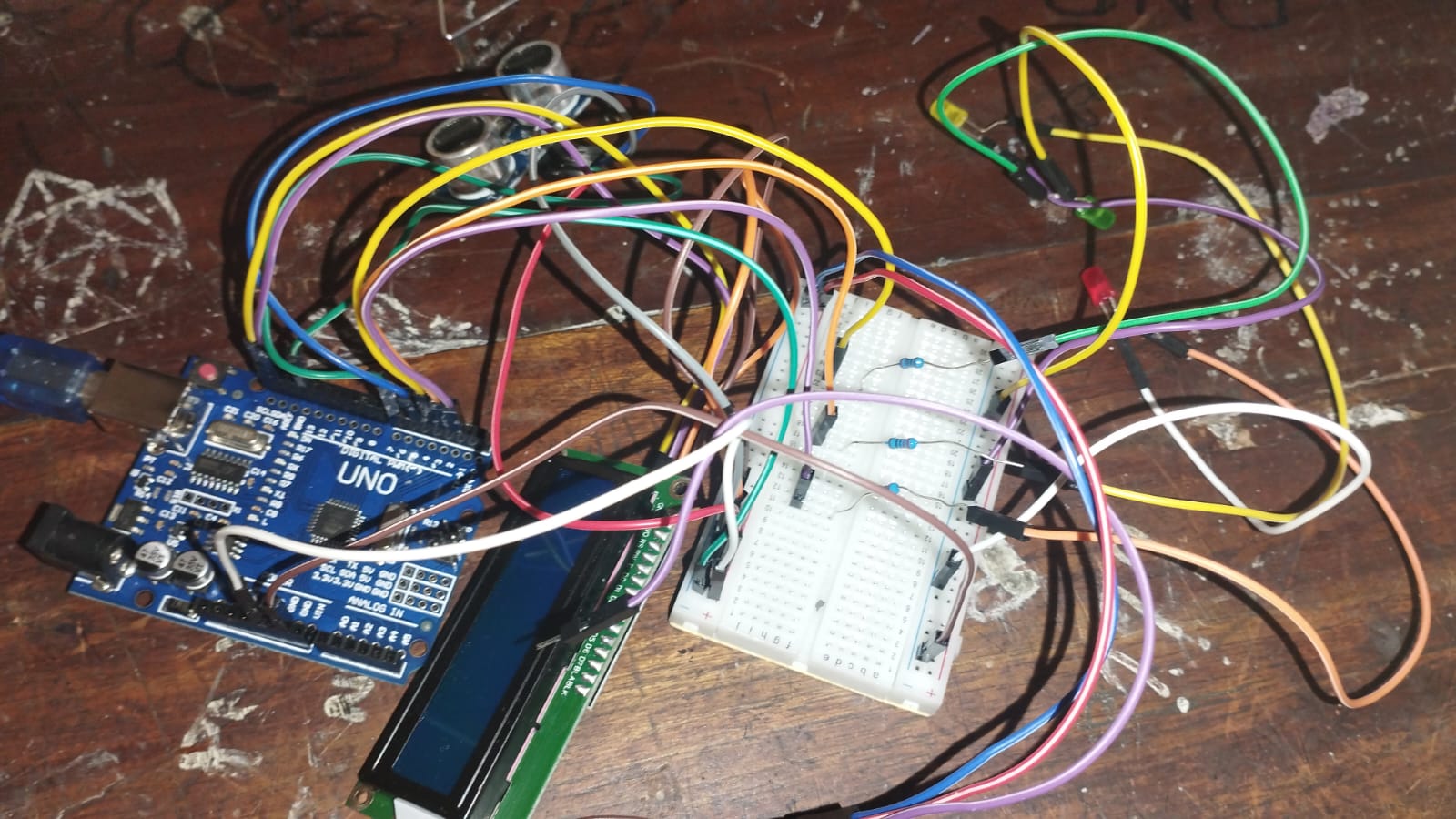


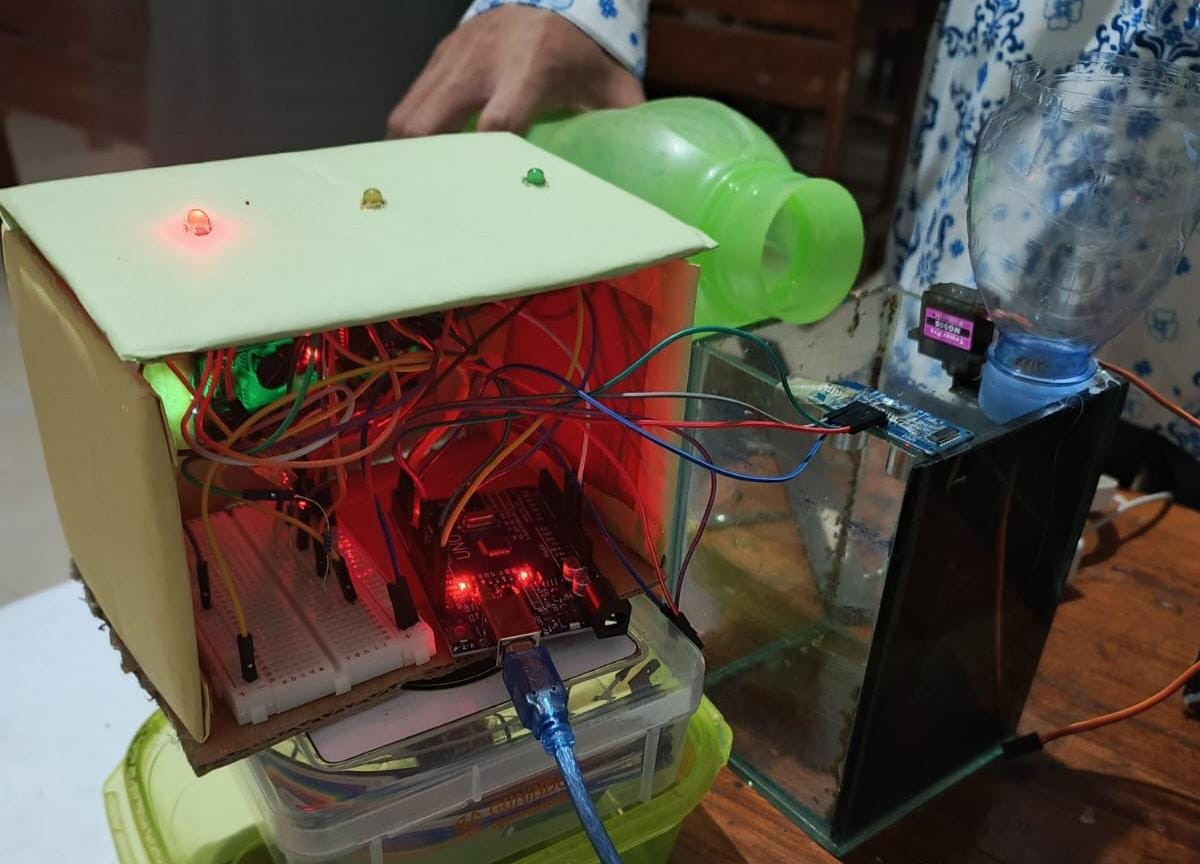
1. **Program**

|  |
| --- |
| #include <Servo.h>  #include <NewPing.h>  #include <Wire.h>  #include <LiquidCrystal\_I2C.h>  const int trigPin = 7;  const int echoPin = 5;  const int maxdistance = 30;  const int servoPin = 9;  const int ledPin\_red = 2;  const int ledPin\_yellow = 3;  const int ledPin\_green = 4;  unsigned long waktu\_servosebelumnya = 0;  unsigned long interval\_waktuservo = 10000;  Servo myservo;  NewPing sonar(trigPin, echoPin, maxdistance);  LiquidCrystal\_I2C lcd(0x27, 16, 2);  void setup() {  myservo.attach(servoPin);  myservo.write(0);  Serial.begin(9600);  pinMode(ledPin\_red, OUTPUT);  pinMode(ledPin\_yellow, OUTPUT);  pinMode(ledPin\_green, OUTPUT);  lcd.begin(16, 2);  lcd.backlight();  lcd.clear();  }  void loop() {  unsigned long waktusekarang = millis();  if (waktusekarang - waktu\_servosebelumnya >= interval\_waktuservo) {  myservo.write(90);  delay(2000);  myservo.write(0);  waktu\_servosebelumnya = waktusekarang;  }  int distance = sonar.ping\_cm();  int ketinggian = maxdistance - distance;  Serial.print("Ketinggian Air: ");  Serial.print(ketinggian);  Serial.println(" cm");  lcd.setCursor(0, 0);  lcd.print("Tinggi Air: ");  lcd.print(ketinggian);  lcd.print("cm ");  if (ketinggian < 5) {  digitalWrite(ledPin\_red, HIGH);  digitalWrite(ledPin\_yellow, LOW);  digitalWrite(ledPin\_green, LOW);  lcd.setCursor(0, 1);  lcd.print("Air Rendah ");  } else if (ketinggian >= 5 && ketinggian < 9) {  digitalWrite(ledPin\_red, LOW);  digitalWrite(ledPin\_yellow, HIGH);  digitalWrite(ledPin\_green, LOW);  lcd.setCursor(0, 1);  lcd.print("Air Sedang ");  } else {  digitalWrite(ledPin\_red, LOW);  digitalWrite(ledPin\_yellow, LOW);  digitalWrite(ledPin\_green, HIGH);  lcd.setCursor(0, 1);  lcd.print("Air Tinggi ");  }  delay(500);  } |

**TAHAP 3 – IMPLEMENTASI**

1. **Dokumentasi**







1. **Kondisi 1**

**Input dan Output Ketika Ketinggian Air < 5 cm**

* **Input**:
  + Data dari sensor ultrasonik menunjukkan jarak antara sensor dengan air lebih besar dari 8 cm (jarak air rendah).
* **Proses**:
  + Nilai jarak dikonversi menjadi nilai ketinggian air, kemudian dibandingkan dengan batas-batas yang telah ditentukan.
* **Output**:
  + LED merah menyala.
  + LCD menampilkan pesan "Air Rendah".



1. **Kondisi 2**

**Input dan Output Ketika Ketinggian Air 5 - 9 cm**

* **Input**:
  + Data dari sensor ultrasonik menunjukkan jarak air berada di rentang 5 – 9 cm.
* **Proses**:
  + Nilai jarak dikonversi menjadi nilai ketinggian air, lalu dikategorikan sebagai kondisi "Air Sedang".
* **Output**:
  + LED kuning menyala.
  + LCD menampilkan pesan "Air Sedang".



1. **Kondisi 3**

**Input dan Output Ketika Ketinggian Air >= 9 cm**

* **Input**:
  + Data dari sensor ultrasonik menunjukkan jarak air kurang dari 5 cm (air tinggi).
* **Proses**:
  + Nilai jarak dikonversi menjadi nilai ketinggian air, lalu dikategorikan sebagai kondisi "Air Tinggi".
* **Output**:
  + LED hijau menyala.
  + LCD menampilkan pesan "Air Tinggi".



1. **Kondisi 4**

**Pemberian Pakan Otomatis**

* **Kondisi:**
  + **Sistem otomatis menggerakkan servo setiap 10 detik untuk membuka pintu pakan ikan.**
* **Output:**
  + **Servo membuka pintu selama 2 detik untuk mengeluarkan pakan, lalu kembali menutup.**

****

**KESIMPULAN**

1. **Kesimpulan**

Prototype **Sistem Otomatisasi Peternakan Ikan** yang telah dirancang dan diuji berhasil memenuhi sebagian besar tujuan utamanya. Alat ini mampu memberikan pakan secara otomatis sesuai dengan interval waktu yang telah ditentukan dan mengatur pengkondisian berdasarkan ketinggian air pada kolam atau wadah. Meskipun masih dalam tahap pengembangan dan terbatas pada penggunaan di akuarium kecil, alat ini menunjukkan potensi besar sebagai solusi inovatif untuk membantu peternak ikan mengelola peternakan mereka dengan lebih mudah dan efisien. Dengan implementasi dan pengembangan yang lebih baik, alat ini diharapkan dapat menjadi bagian penting dalam mendukung transformasi peternakan berbasis teknologi modern.

1. **Kendala**

Beberapa tantangan teknis dihadapi selama pengembangan alat ini. Kendala pertama adalah penggunaan fungsi delay yang menyebabkan terganggunya fungsi monitoring saat servo aktif. Masalah ini berhasil diselesaikan dengan mengganti fungsi delay menjadi fungsi millis, yang memungkinkan sistem untuk berjalan secara paralel tanpa gangguan. Kendala kedua adalah keterbatasan daya yang mengakibatkan ketidakmampuan alat untuk menjalankan semua komponen secara bersamaan, sehingga komponen pompa air harus diganti dengan LED sebagai indikator pengkondisian.

1. **Saran**

Untuk pengembangan lebih lanjut, disarankan untuk menggunakan power supply dengan kapasitas daya dan tegangan yang lebih besar agar semua komponen dapat berfungsi secara optimal. Penambahan modul IoT dapat memberikan kemampuan kendali jarak jauh, sehingga meningkatkan fleksibilitas alat. Selain itu, integrasi pompa air yang lebih sesuai, desain sistem yang tahan terhadap kondisi lingkungan, dan penambahan modul RTC untuk penyesuaian waktu yang lebih presisi juga sangat direkomendasikan.