

# LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Praktikum Implementasi Simulasi Sensor Jarak Ultrasonik menggunakan  
ESP32 melalui IoT**



*Dafa Putra Ady Pratama*  
*Fakultas Vokasi, Universitas Brawijaya*  
*Email [daffap1499@gmail.com](mailto:daffap1499@gmail.com)*

## Abstract

Praktikum ini bertujuan untuk mengimplementasikan simulasi sensor jarak ultrasonik menggunakan mikrokontroler ESP32 yang terintegrasi dengan Internet of Things (IoT). Sensor ultrasonik digunakan untuk mendeteksi jarak suatu objek dengan mengukur waktu tempuh gelombang ultrasonik dari sensor ke objek dan kembali ke sensor. ESP32 berperan sebagai pengendali utama yang mengolah data dari sensor dan mengirimkannya ke platform IoT untuk pemantauan jarak secara real-time. Data yang diperoleh ditampilkan melalui antarmuka web atau aplikasi seluler, memungkinkan pengguna memantau hasil pengukuran secara fleksibel. Hasil praktikum menunjukkan bahwa sistem ini mampu mendeteksi jarak dengan akurasi yang baik dan memberikan respons cepat dalam pengiriman data ke platform IoT. Implementasi ini diharapkan menjadi dasar untuk pengembangan sistem pemantauan jarak yang lebih kompleks dan aplikasi lain yang berbasis IoT..

**Keywords**—*Internet of Things, ESP32, HC-SR04*

## 1. Introduction

### 1.1 Latar Belakang Praktikum IoT yang Dilakukan

Perkembangan teknologi Internet of Things (IoT) membuka peluang untuk menghubungkan perangkat fisik dengan internet, termasuk dalam pemantauan jarak menggunakan sensor ultrasonik. ESP32, dengan koneksi Wi-Fi dan Bluetooth, memudahkan pengiriman data secara nirkabel ke platform IoT. Praktikum ini dilakukan untuk mempelajari integrasi sensor ultrasonik dengan ESP32 serta proses pengiriman data jarak secara real-time, yang menjadi dasar dalam pengembangan sistem berbasis IoT.

### 1.2 Tujuan Eksperimen

Tujuan dari eksperimen ini adalah untuk mempelajari dan mengimplementasikan integrasi sensor ultrasonik dengan mikrokontroler ESP32 dalam sistem berbasis Internet of Things (IoT). Melalui eksperimen ini, diharapkan dapat dipahami proses pengukuran jarak secara real-time serta pengiriman data secara nirkabel ke platform IoT untuk mendukung pemantauan jarak yang efisien dan akurat.

## 2. Methodology

### 2.1 Tools & Materials

- **Mikrokontroler:** ESP32
- **HC-SR04:** 1 buah HC-SR04
- **Software:** Visual Code, Web Browser
- **Koneksi Jaringan:** Wi-Fi

### 2.2 Implementation Steps

1. **Persiapan Hardware:**
  - Hubungkan HC-SR04 dengan ESP32
2. **Pengaturan Software:**
  - Program ESP32 menggunakan Arduino IDE.
  - Konfigurasi Wi-Fi untuk menghubungkan ESP32 ke jaringan internet.
  - Gunakan server web lokal pada ESP32 untuk mengontrol HC-SR04.

### 3. Pengkodean dan Implementasi Web:

- Tulis kode untuk mengendalikan ESP32, yang akan merespon perintah dari halaman web.
- Program untuk mengubah status ESP32

### 4. Pengujian:

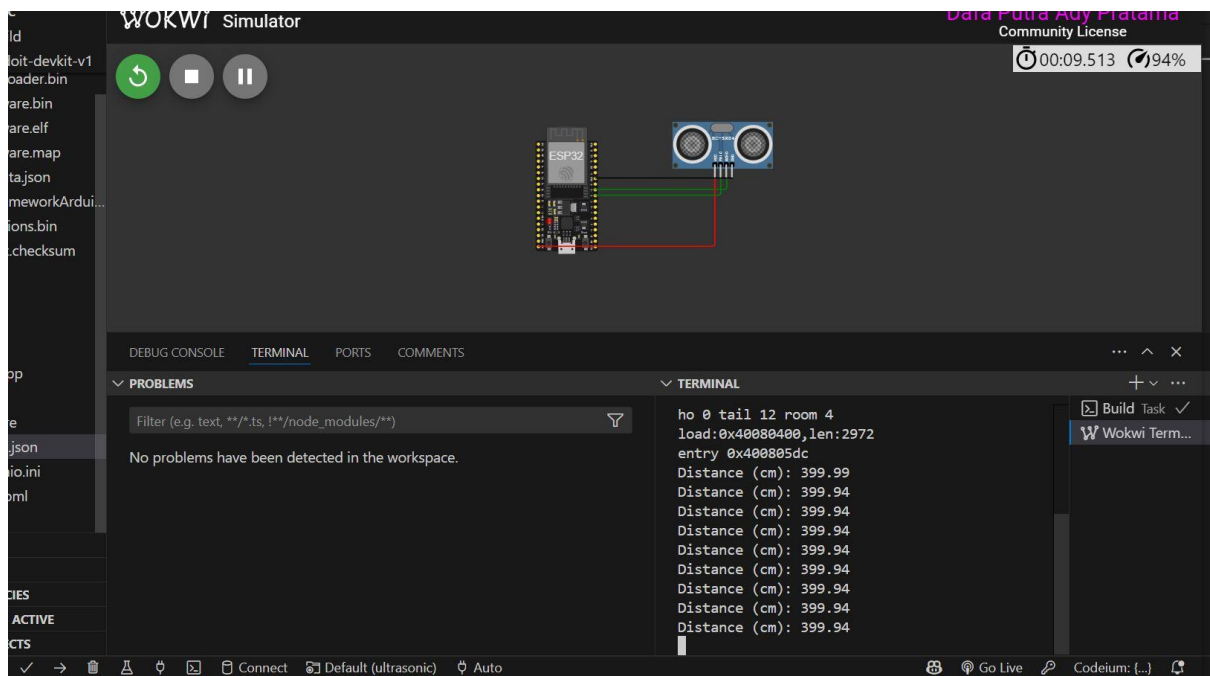
- Uji sistem dengan mengakses halaman web dan mengontrol gelombang HC-SR04.

## 3. Results and Discussion

### 3.1 Experimental Results

Pada eksperimen ini, sensor ultrasonik HC-SR04 dihubungkan dengan mikrokontroler ESP32 untuk mengukur jarak objek secara real-time. Data hasil pengukuran dikirimkan ke perangkat yang terhubung melalui koneksi nirkabel menggunakan fitur Wi-Fi pada ESP32. Dari hasil pengujian, sistem berhasil mendeteksi jarak dengan akurasi yang konsisten dalam rentang 2 hingga 400 cm. Pengolahan data oleh ESP32 menunjukkan respons yang cepat, dengan hasil pengukuran yang stabil dan ditampilkan secara real-time. Performa sistem dipengaruhi oleh faktor seperti sudut deteksi dan permukaan objek, namun secara keseluruhan, integrasi HC-SR04 dengan ESP32 berjalan dengan baik untuk aplikasi pemantauan jarak berbasis IoT.

Berikut adalah hasilnya:



## 4. Appendix

```
1  #include <Arduino.h>
2
3  const int trigPin = 5;
4  const int echoPin = 18;
5
6
7  //define sound speed in cm/uS
8  #define SOUND_SPEED 0.034
9  #define CM_TO_INCH 0.393701
10
11
12  long duration;
13  float distanceCm;
14  float distanceInch;
15
16
17  void setup() {
18    Serial.begin(115200); // Starts the serial communication
19    pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
20    pinMode(echoPin, INPUT); // Sets the echoPin as an Input
21  }
22
23
24  void loop() {
25    // Clears the trigPin
26    digitalWrite(trigPin, LOW);
27    delayMicroseconds(2);
28    // Sets the trigPin on HIGH state for 10 micro seconds
29    digitalWrite(trigPin, HIGH);
30    delayMicroseconds(10);
31    digitalWrite(trigPin, LOW);
32    // Reads the echoPin, returns the sound wave travel time in microseconds
33    duration = pulseIn(echoPin, HIGH);
34    // Calculate the distance
35    distanceCm = duration * SOUND_SPEED/2;
36    // Convert to inches
37    distanceInch = distanceCm * CM_TO_INCH;
38    // Prints the distance in the Serial Monitor
39    Serial.print("Distance (cm): ");
40    Serial.println(distanceCm);
41    // Serial.print("Distance (inch): ");
42    // Serial.println(distanceInch);
43    delay(1000);
44  }
45
```

