# LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Praktik Simulasi Sensor Jarak (Ultrasonic)**



*Fawwaz Mufid Wardaya*

*Fakultas Vokasi, Universitas Brawijaya*

*Email : mahesfawwaz79@gmail.com*

**Abstract**

Tujuan dari eksperimen ini adalah untuk mempelajari cara kerja sensor ultrasonik (HC-SR04), yang terhubung ke mikrokontroler ESP32 dan dapat mendeteksi dan mengukur jarak objek secara real-time. Sensor ini bekerja dengan prinsip pantulan gelombang suara ultrasonik untuk mendeteksi objek dan mengukur jarak. Selain itu, eksperimen ini akan menunjukkan beberapa aplikasi IoT yang dapat memantau jarak, seperti sistem parkir otomatis, detergent, dan mesin cuci. Untuk analisis lebih lanjut, hasil pembacaan jarak ditampilkan pada Serial Monitor.

**1. Introduction**

**1.1 Latar belakang**

Ada banyak aplikasi cerdas yang dapat membantu manusia dalam kehidupan sehari-hari berkat Internet of Things (IoT). Sensor adalah bagian penting dari Internet of Things karena mereka berfungsi untuk mendeteksi perubahan dalam lingkungan sekitar mereka. Sensor ultrasonik (HC-SR04) mengirimkan gelombang ultrasonik dan mengukur waktu yang diperlukan gelombang tersebut untuk kembali setelah memantul dari objek. Ini adalah salah satu sensor yang paling sering digunakan untuk mengukur jarak dengan akurasi tinggi tanpa bersentuhan fisik.

Tujuan dari praktik ini adalah untuk mempelajari dasar-dasar penggunaan sensor ultrasonik dalam aplikasi IoT, seperti sistem parkir otomatis atau deteksi objek industri. Kami menggunakan ESP32 sebagai mikrokontroler utama untuk membaca data dari sensor ultrasonik dan menampilkannya pada Serial Monitor.

**1.2 Tujuan eksperimen**

Tujuan dari tes ini adalah:

1. Mempelajari cara kerja sensor ultrasonik HC-SR04 dan integrasinya dengan ESP32.
2. Mengukur jarak objek secara real-time menggunakan sensor ultrasonik.
3. Menunjukkan potensi aplikasi IoT dalam pemantauan jarak, seperti sistem parkir otomatis atau deteksi objek.

**2. Methodology (Metodologi)**

**2.1 Tools & Materials (Alat dan Bahan)**

* Mikrokontroler : ESP32
* Sensor Ultrasonik: HC-SR04
* Kabel Jumper
* Software : VSCODE

**2.2 Implementation Steps (Langkah Implementasi)**

1. **Persiapan Perangkat :**

* Pin VCC HC-SR04 → 5V ESP32
* Pin GND HC-SR04 → GND ESP32
* Pin Trig HC-SR04 → Pin GPIO tertentu, misalnya GPIO 5
* Pin Echo HC-SR04 → Pin GPIO tertentu, misalnya GPIO 18

1. **Code :**

* Buka VSCode dan instal library yang diperlukan (jika ada).
* Tuliskan Code Program :

1. Mengirimkan sinyal trigger ke sensor ultrasonik.
2. Membaca sinyal echo untuk menghitung waktu tempuh gelombang ultrasonik.
3. Menghitung jarak berdasarkan waktu tempuh gelombang ultrasonik.
4. **Pengujian :**

* Upload ke ESP32.
* Uji sistem dengan menempatkan objek pada jarak tertentu dari sensor ultrasonik. Amati hasil pembacaan jarak di Serial Monitor.

**3. Results and Discussion (Hasil dan Pembahasan)**

**3.1 Experimental Results (Hasil Eksperimen)**

Setelah kode diupload ke ESP32, sensor ultrasonik berhasil membaca data jarak objek secara real-time. Berikut hasil pengujian:

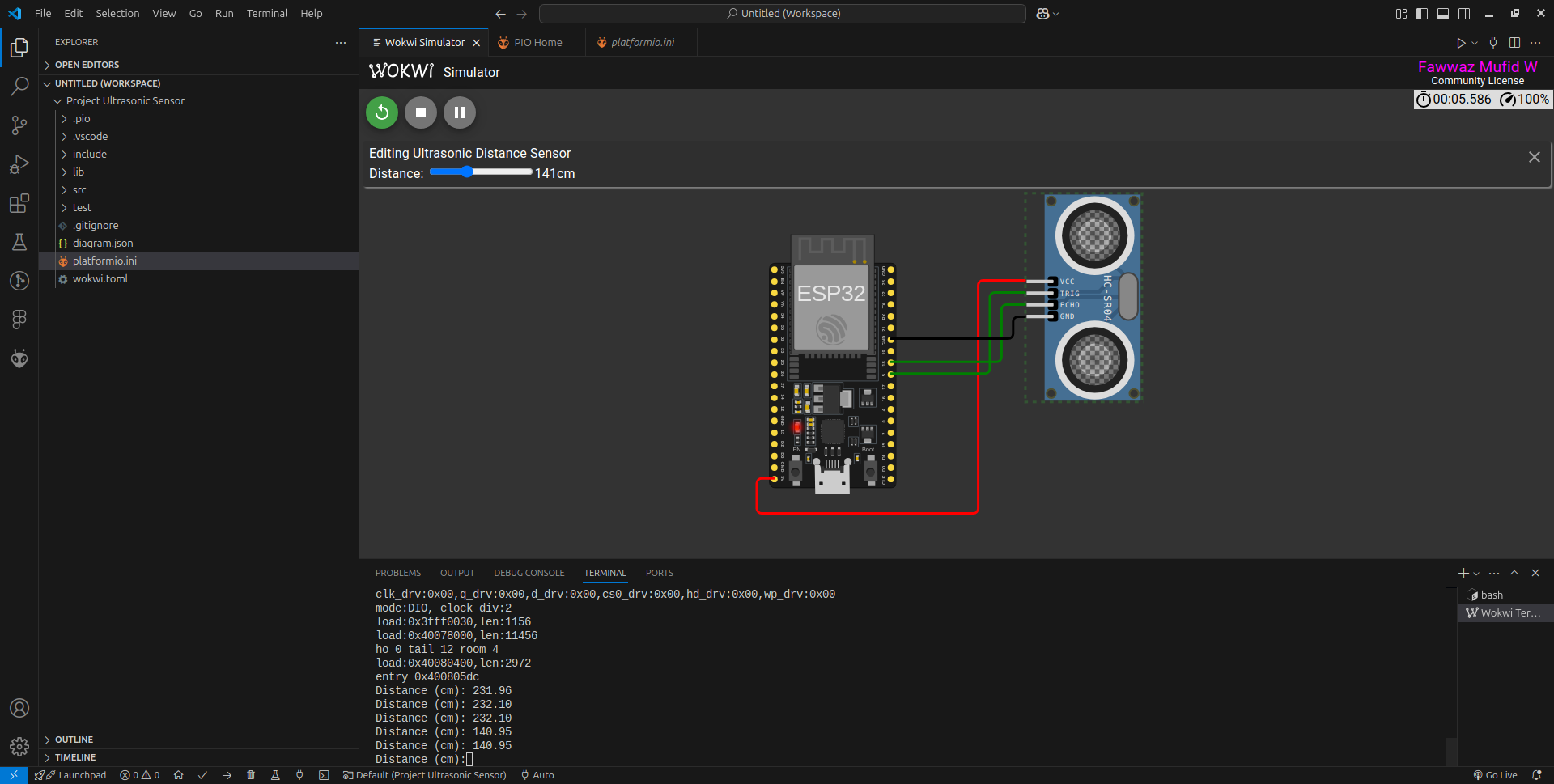
Distance (cm): 399.98

Distance (cm): 399.94

Distance (cm): 399.94

Distance (cm): 232.10

Distance (cm): 232.10



**4. Appendix (Lampiran, jika diperlukan)**

**Kode Program :**

#include <Arduino.h>

const int trigPin = 5;

const int echoPin = 18;

//define sound speed in cm/uS

#define SOUND\_SPEED 0.034

#define CM\_TO\_INCH 0.393701

long duration;

float distanceCm;

float distanceInch;

void setup() {

Serial.begin(9600); // Starts the serial communication

pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output

pinMode(echoPin, INPUT); // Sets the echoPin as an Input

}

void loop() {

// Clears the trigPin

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

// Sets the trigPin on HIGH state for 10 micro seconds

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

// Reads the echoPin, returns the sound wave travel time in microseconds

duration = pulseIn(echoPin, HIGH);

// Calculate the distance

distanceCm = duration \* SOUND\_SPEED/2;

// Convert to inches

distanceInch = distanceCm \* CM\_TO\_INCH;

// Prints the distance in the Serial Monitor

Serial.print("Distance (cm): ");

Serial.println(distanceCm);

// Serial.print("Distance (inch): ");

// Serial.println(distanceInch);

delay(1000);

}