# LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Praktik Simulasi Sensor Jarak**

**(Ultrasonic)**

*Eva Latifah*

*Fakultas Vokasi, Universitas Brawijaya*

*Email: ev.tifa@gmail.com*

**Abstrak**

Praktikum ini bertujuan untuk mengukur jarak menggunakan sensor ultrasonik berbasis modul HC-SR04 yang dikendalikan oleh mikrokontroler ESP32. Sistem dikembangkan menggunakan bahasa pemrograman C++ dalam lingkungan Arduino IDE. Data jarak diukur berdasarkan waktu pantulan gelombang ultrasonik dan dikonversi menjadi satuan sentimeter dan inci. Hasil menunjukkan bahwa ESP32 mampu membaca dan menampilkan jarak dengan akurasi yang baik melalui komunikasi serial. Implementasi ini dapat dikembangkan lebih lanjut untuk berbagai aplikasi seperti sistem parkir otomatis, robotika, dan perangkat IoT.

**Kata Kunci**—*ESP32, Sensor Ultrasonik, HC-SR04, Jarak, Arduino IDE.*

***Abstract***

*This practicum aims to measure distance using an ultrasonic sensor based on the HC-SR04 module, controlled by an ESP32 microcontroller. The system is developed using the C++ programming language within the Arduino IDE environment. Distance data is measured based on the reflection time of ultrasonic waves and converted into centimeters and inches. The results show that the ESP32 can accurately read and display distance through serial communication. This implementation can be further developed for various applications such as automatic parking systems, robotics, and IoT devices.*

***Keywords****— ESP32, Ultrasonic Sensor, HC-SR04, Distance, Arduino IDE.*

**1. Introduction**

**1.1 Latar belakang**

Sensor ultrasonik merupakan perangkat yang sering digunakan untuk mendeteksi objek dan mengukur jarak tanpa kontak fisik. Prinsip kerja sensor ini didasarkan pada pemantulan gelombang ultrasonik yang dikirimkan dan diterima kembali setelah mengenai suatu objek. Salah satu sensor ultrasonik yang umum digunakan adalah HC-SR04, yang dapat mengukur jarak dengan akurasi tinggi hingga beberapa meter.

Dalam eksperimen ini, sensor HC-SR04 dihubungkan dengan mikrokontroler ESP32 untuk melakukan pengukuran jarak secara real-time. Data yang diperoleh dikirimkan melalui komunikasi serial dan ditampilkan pada Serial Monitor. Implementasi ini sangat berguna untuk berbagai aplikasi, termasuk sistem keamanan, robotika, dan kendaraan otonom.

**1.2 Tujuan eksperimen**

Praktikum ini bertujuan untuk:

1. Memahami cara kerja sensor ultrasonik dalam mengukur jarak.
2. Menghubungkan sensor HC-SR04 dengan ESP32 dan membaca data melalui komunikasi serial.
3. Mengonversi data jarak ke dalam satuan sentimeter dan inci.

**2. Methodology**

**2.1 Tools & Materials**

1. Mikrokontroler ESP32
2. Sensor Ultrasonik HC-SR04
3. Breadboard dan kabel jumper
4. Software Arduino IDE
5. Simulator Wokwi

**2.2 Implementation Steps**

1. Koneksi Hardware:
   1. VCC → 3.3V ESP32
   2. GND → GND ESP32
   3. TrigPin → GPIO 5 ESP32
   4. EchoPin → GPIO 18 ESP32
2. Pemrograman dan Pengujian:
   1. Mengunggah kode program ke ESP32 menggunakan Arduino IDE.
   2. Menjalankan program dan membaca data jarak melalui Serial Monitor.
   3. Menganalisis keakuratan pengukuran berdasarkan nilai yang ditampilkan.

**3. Results and Discussion**

**3.1 Experimental Results**

Pada eksperimen ini, ESP32 digunakan untuk mengontrol sensor HC-SR04 guna mengukur jarak objek di depannya. Sensor bekerja dengan mengirimkan gelombang ultrasonik melalui pin **Trig**, kemudian menerima pantulan gelombang melalui pin **Echo**. Waktu yang dibutuhkan gelombang untuk kembali digunakan untuk menghitung jarak dengan rumus berikut:

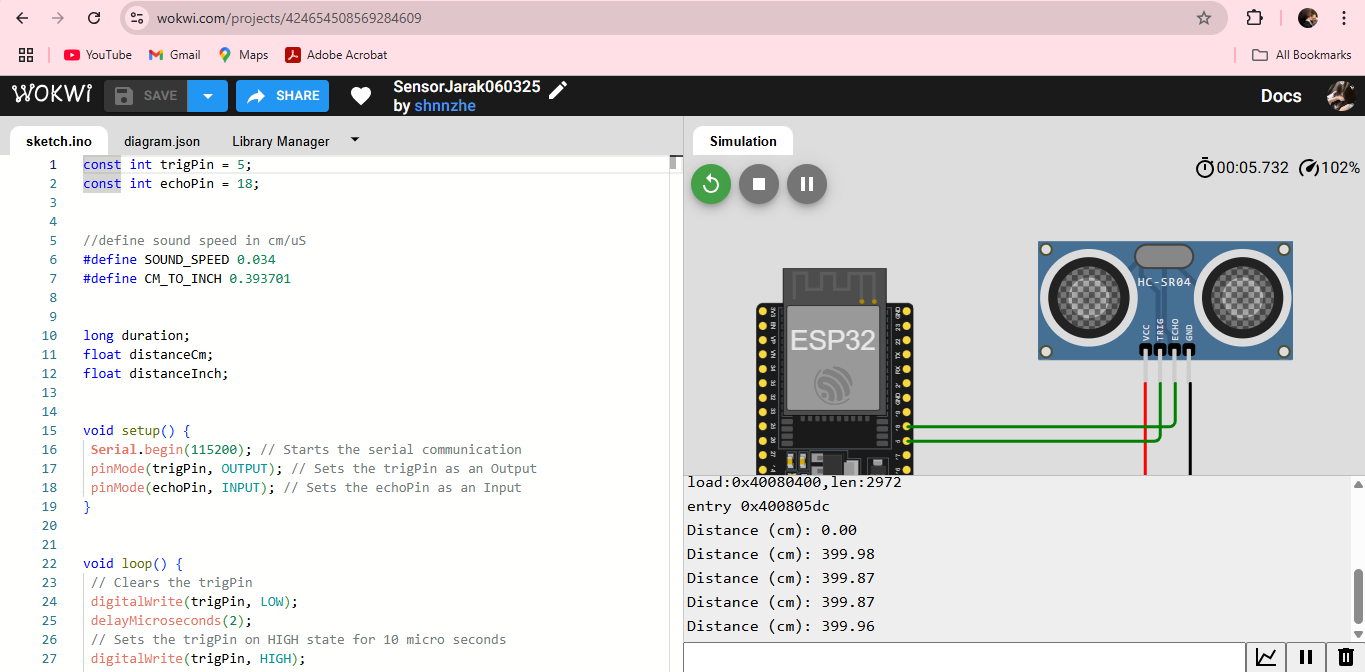
Jarak = Waktu Pantulan x Kecepatan Suara/2

Di mana kecepatan suara dalam udara adalah **0.034 cm/µs**. Hasil pengukuran ditampilkan dalam satuan sentimeter dan inci. Dari pengujian yang dilakukan, sistem mampu mengukur jarak dengan akurasi yang cukup baik dan menampilkan hasil dalam waktu nyata pada Serial Monitor.

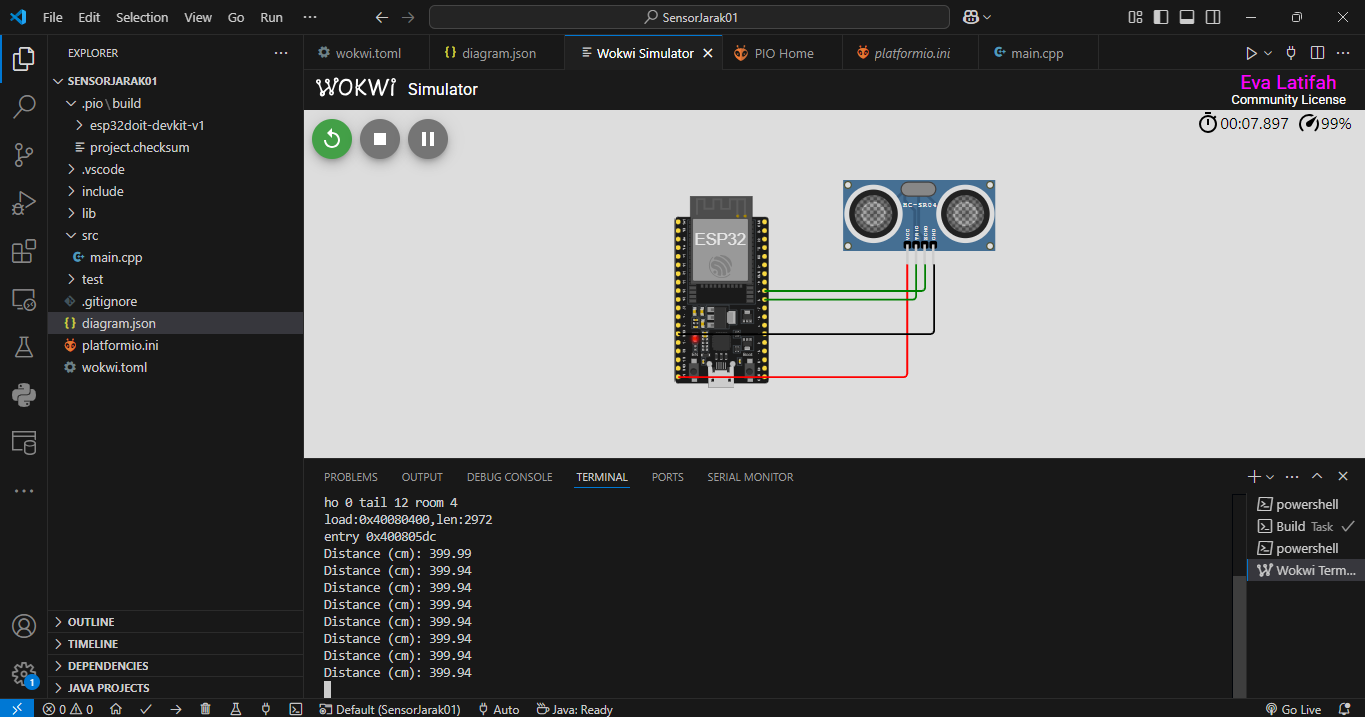
Keunggulan utama sistem ini adalah respons cepat dan kemudahan implementasi. Namun, terdapat beberapa faktor yang dapat mempengaruhi akurasi pengukuran, seperti sudut objek, permukaan reflektif, serta gangguan lingkungan seperti suhu dan kelembaban.

Untuk meningkatkan keandalan sistem, dapat dilakukan perbaikan dengan menambahkan **filter median** untuk mengurangi noise atau **penyesuaian kalibrasi sensor** agar hasil pengukuran lebih akurat. Selain itu, pengembangan lebih lanjut dapat mencakup **komunikasi berbasis IoT** untuk pemantauan jarak jauh.

**Result Wokwi:**



**Result Vscode:**

****

**4. Appendix**

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(115200); // 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);

}