

# LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

(Praktik Simulasi Sensor Jarak Berbasis IoT dengan ESP32 Menggunakan Platform  
Wokwi yang Diintegrasikan dengan VS Code)



*Muhammad Kadavi*  
*Fakultas Vokasi, Universitas Brawijaya*  
*Email: [kadavi2945@student.ub.ac.id](mailto:kadavi2945@student.ub.ac.id)*

## Abstrak

Praktikum ini mensimulasikan sistem sensor jarak HC-SR04 berbasis IoT dengan menggunakan ESP32 pada platform Wokwi yang diintegrasikan dengan Visual Studio Code (VS Code) melalui ekstensi PlatformIO. Sensor jarak dan diuji untuk mendeteksi objek secara virtual dimana data hasil pengukuran ditampilkan secara real-time melalui antarmuka Wokwi dan terminal vscode. Pada sistem ini, ESP32 berperan sebagai penggerak utama, yang memproses sinyal pengukuran sensor dan mengirimkannya kembali melalui antarmuka logika pemrograman yang dikembangkan di VS Code. Hasil simulasi menunjukkan sistem mampu merespon perubahan jarak dengan stabil. Membuktikan bahwa konsep integrasi Wokwi dan VS Code efektif untuk pengembangan dan debugging dispositif IoT simpel tanpa diperlukannya hardware fisik. Secara konseptual, simulasi ini dapat dijadikan model dasar untuk implementasi sistem ATSM dalam aplikasi smart automation.

## Pendahuluan

Pertumbuhan IoT yang sama menimbulkan permintaan atas simulasi komponen elektronik berbasis mikrokontroler seperti ESP32 untuk pembelajaran yang efektif, dan biaya-efektif. Praktik kerja ini membangun sistem sensor berbasis IoT – terutama, sensor jarak – dari nol, menggunakan ESP32 sebagai perekaman data sentral. Simulasi ini memungkinkan modul sensor HC-SR04 mengirimkan data virtual ke Visual Studio Code (VS Code). Praktik kerja ini memungkinkan untuk merancang, mendeploy, dan memonitor sistem sensor IoT dengan pembatasan real-time dan biaya minim.

## Latar Belakang

Permasalahan peningkatan kebutuhan sistem IoT dalam kehidupan modern harus direspons dengan metode pembelajaran yang praktis dan hemat biaya. Simulasi komponen elektronik, pada kasus ini sensor jarak dengan ESP32, menjadi solusi efektif untuk pemahaman konsep IoT tanpa perlu bergantung pada keberadaan perangkat fisik. Dengan aplikasi platform Wokwi pada Visual Studio Code, pengembangan dan pengujian sebagian logika pemrograman dapat digelar secara virtual, yang mencakup simulasi respons dari sensor yang ditempatkan terhadap objek, visualisasi data dalam waktu nyata, serta debugging yang lebih cepat dan efisien. Keseluruhan praktik bertujuan membangun fundasi pemahaman integrasi sensor-aktor dalam IoT berbasis kondisi nyata untuk mempersiapkan user dalam pengembangan aplikasi otomasi cerdas.

## Tujuan

1. Memahami simulasi sistem sensor jarak berbasis IoT menggunakan ESP32 di Platform Wokwi.
2. Menguji respons sensor terhadap perubahan jarak objek
3. Platform Wokwi dengan VS Code untuk pemrograman dan debugging yang efisien.
4. Membangun dasar pemahaman interaksi sensor-aktor dalam sistem IoT untuk aplikasi otomasi nyata.

## Metodologi

Simulasi sensor jarak berbasis IoT dengan ESP32 dilakukan menggunakan Platform Wokwi yang terintegrasi ke Visual Studio Code melalui ekstensi PlatformIO. Sensor jarak HC-SR04 diprogram untuk mendeteksi objek. Logika dan algoritma ditulis di VS Code untuk membaca data sensor, mengonversi nilai jarak, dan mengaktifkan output sesuai kondisi yang ditentukan. Simulasi dijalankan di Wokwi untuk menguji respons sistem secara real-time, seperti nyala/mati LED atau aktivasi relay saat jarak objek berubah. Validasi dilakukan melalui visualisasi antarmuka Wokwi dan pemantauan data serial untuk memastikan akurasi sistem serta konsistensi algoritma pemrograman.

## Software & Hardware

Software	Hardware
Wokwi	Esp32 (virtual)
Extension PlatformIO	Sensor Jarak HC-SR04 (Virtual)
VSCode	Laptop

## 2.2 Implementation Steps (Langkah Implementasi)

### 1. Persiapan

1. Buka Wokwi (<https://wokwi.com>)
2. Buat proyek baru dengan ESP32, lalu tambahkan komponen:
  - 3 Sensor Jarak HC-SR04

3. **Hubungkan komponen virtual ke ESP32** menggunakan diagram JSON.

## 2. Instalasi VS Code dan PlatformIO

1. **Download dan Install VS Code** dari <https://code.visualstudio.com/>
2. **Buka VS Code**, lalu buka **Extensions Marketplace** (Ctrl+Shift+X).
3. **Cari "PlatformIO"**, lalu **install extension PlatformIO**.
4. **Restart VS Code** setelah instalasi selesai.

## 3. Membuat Proyek di PlatformIO

1. **Buka VS Code** dan **klik ikon PlatformIO** (di sidebar kiri).
2. **Buat proyek baru** dengan:
  - **Board:** DOIT ESP32 DEVKIT V1
  - **Framework:** Arduino
  - **Nama Proyek:** SENSOR\_JARAK

## 4. Buat File wokwi.toml, dan salin syntax di bawah ini

```
[wokwi]
version = 1
firmware = '.pio/build/esp32doit-devkit-
v1/firmware.bin'
elf = '.pio/build/esp32doit-devkit-
v1/firmware.elf'
```

## 4. Menulis Kode ESP32 di VS Code

1. **Buka file src/main.cpp** dan masukkan kode berikut:

```
#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(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);
}

```

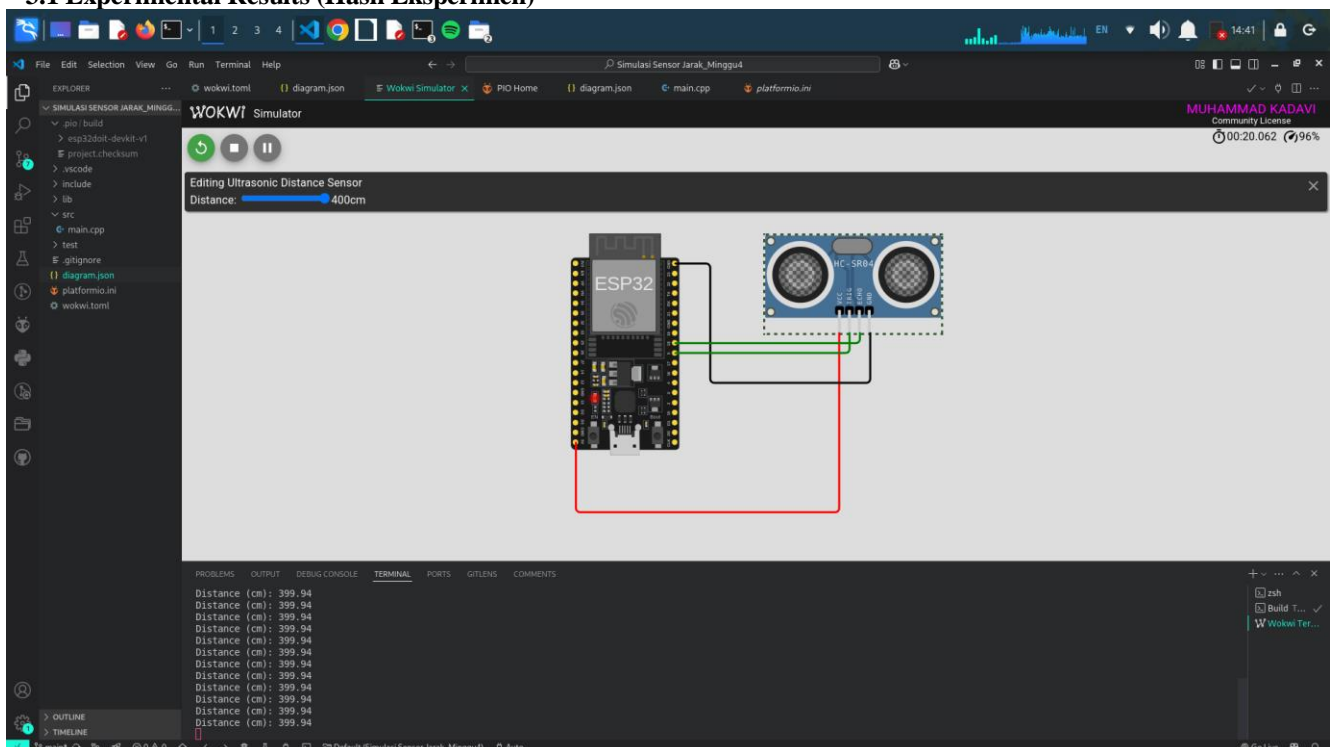
## 5. Menjalankan Simulasi di Wokwi

1. Buka proyek Wokwi yang sudah dibuat.
2. Tambahkan diagram JSON untuk menghubungkan ESP32 dengan LED dan ILI9341.
3. Klik tombol "RUN" untuk menjalankan simulasi dan melihat hasilnya.

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

Simulasi ini dilakukan dan hasilnya menunjukkan bahwa balikan sistem berlangsung real-time tanpa adanya penundaan atau jeda fungsi. Fungsi-fungsi dari komponen sistem tersebut, yaitu sensor HC-SR04, tertulis dan disimulasikan sesuai logika algoritma yang sudah diprogram. Ketika sensor jarak di ubah jaraknya, status dari terminal menunjukkan perubahan data jarak.

### 3.1 Experimental Results (Hasil Eksperimen)



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

```
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
46
```