# LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya



**PRAKTIKUM SIMULASI SENSOR JARAK (ULTRASONIC)**

**Author :**

**Ivan AryaPutra Rachmadhani / 233140700111109**

**Fakultas Vokasi, Universitas Brawijaya**

**Email : ivanarya990@gmail.com**

**Abstract** (Abstrak)

Tujuan dari praktik ini adalah untuk mensimulasikan penggunaan sensor jarak ultrasonik HC-SR04, yang digunakan dengan platform Wokwi dan mikrokontroler ESP32. Sensor ini mengukur jarak suatu objek dengan menggunakan pantulan gelombang ultrasonik yang dihantarkan dan diterima kembali oleh sensor. Ini dapat digunakan dalam berbagai aplikasi berbasis Internet of Things, seperti sistem keamanan, robot navigasi, dan otomatisasi industri.Menggunakan IDE Arduino, kode program dikembangkan untuk simulasi ini untuk mengontrol sensor HC-SR04 dan menampilkan hasil pengukuran jarak pada monitor serial. Dengan platform Wokwi, pengembangan dan optimasi dapat dilakukan dengan mudah tanpa menggunakan perangkat keras fisik.

Hasil praktikum menunjukkan bahwa sensor HC-SR04 dapat mengukur jarak dengan akurasi yang cukup, dan sistem dapat beroperasi dengan baik dalam lingkungan simulasi. Proses debugging dan pengujian sistem sebelum implementasi di dunia nyata telah terbukti lebih mudah dengan platform Wokwi. Oleh karena itu, praktikum ini memberikan pemahaman mendalam tentang penerapan sensor ultrasonik dalam teknologi Internet of Things (IoT) dan menunjukkan bagaimana sensor ini dapat digunakan bersama mikrokontroler ESP32 untuk membuat sistem IoT yang cerdas dan adaptif.

**1. Introduction**

**1.1 Latar belakang**

Sensor jarak ultrasonik adalah salah satu dari banyak sensor yang dapat digunakan dalam berbagai aplikasi berkat kemajuan teknologi Internet of Things (IoT). Sensor ini biasanya digunakan untuk mendeteksi objek, mengukur jarak, dan mengontrol perangkat otomatis seperti kendaraan pintar dan sistem keamanan.Simulasi sensor jarak ultrasonik dilakukan dalam praktikum ini dengan platform Wokwi. Platform ini memungkinkan pengujian sistem tanpa perangkat keras fisik, yang memudahkan proses perancangan dan pengujian sebelum implementasi nyata.

**1.2 Tujuan eksperimen**

1. Mempelajari cara kerja sensor jarak ultrasonik HC-SR04.
2. Mengimplementasikan sensor jarak dalam simulasi menggunakan Wokwi.
3. Mengintegrasikan sensor dengan mikrokontroler ESP32.
4. Mengamati data pengukuran yang dihasilkan sensor jarak dalam simulasi.

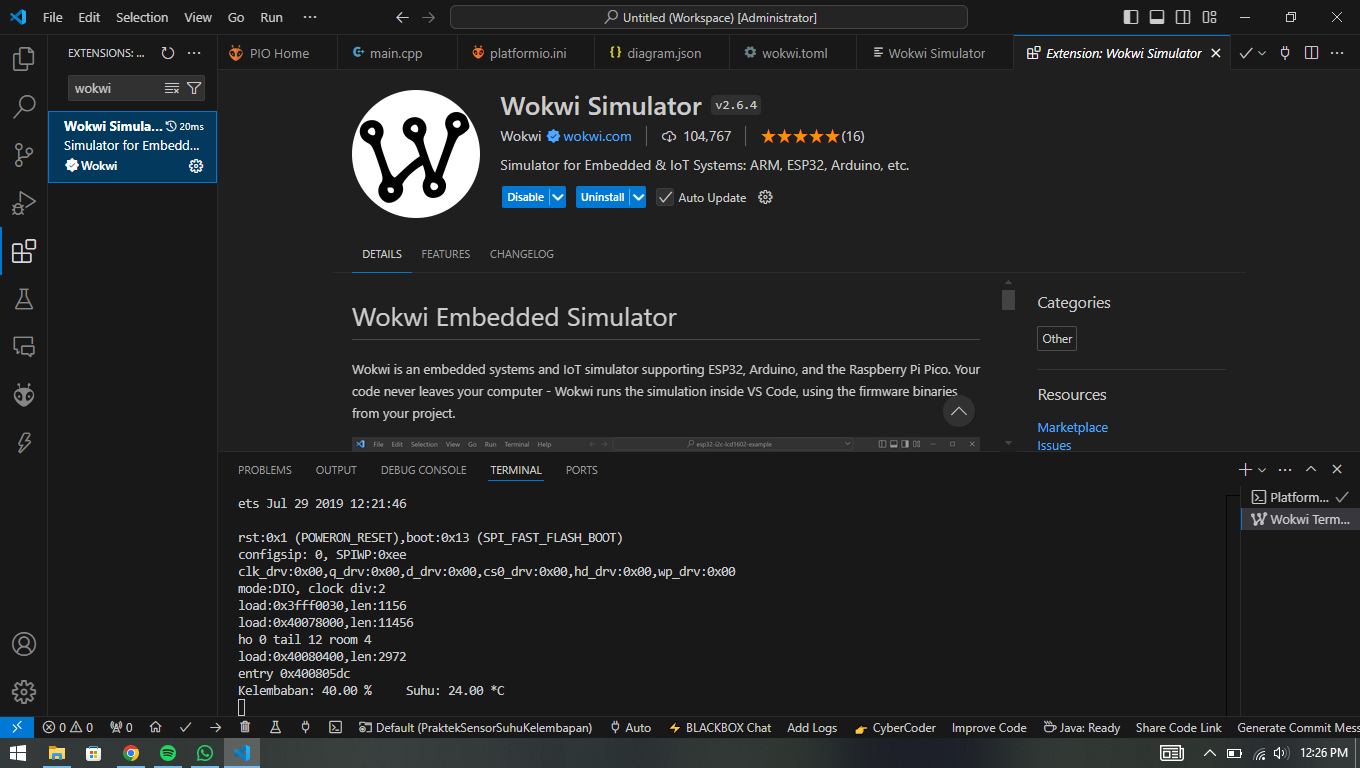
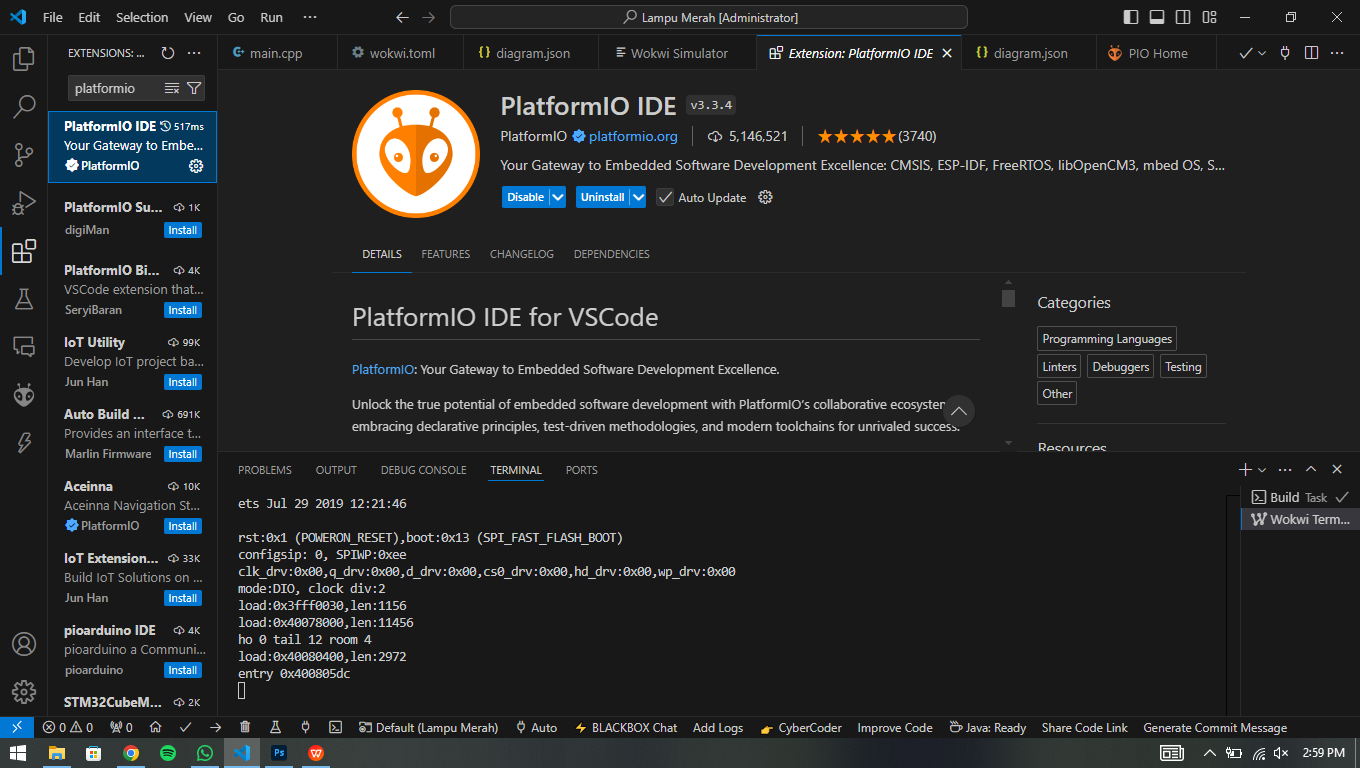
**2. Methodology (Metodologi)**

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

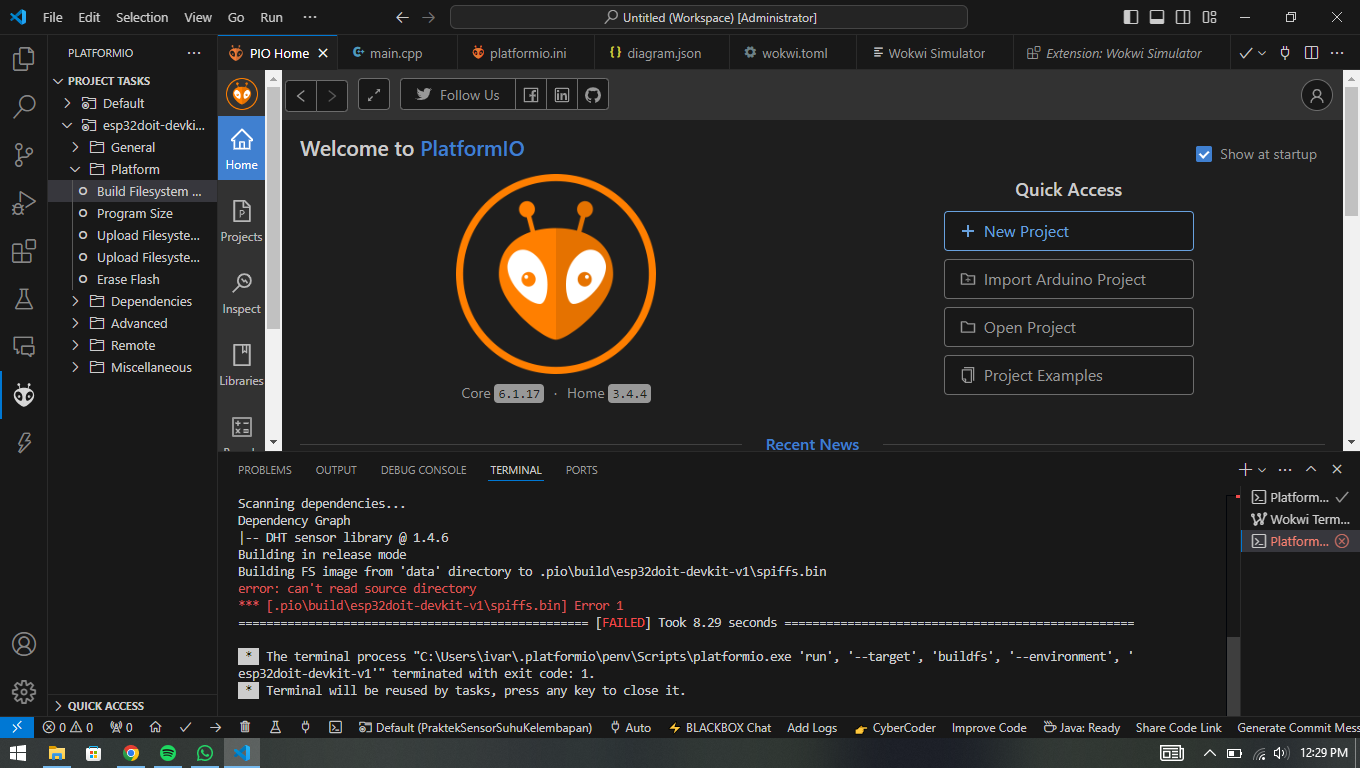
* **Mikrokontroler**: Doit ESP32 Devkit V1
* **Sensor** : HC-SR04 (Ultrasonic Sensor)
* Software : Arduino
* Platfrom : Wokwi, VScod, PlatformIO

**2.2 Implementation Steps (Langkah Implementasi)**

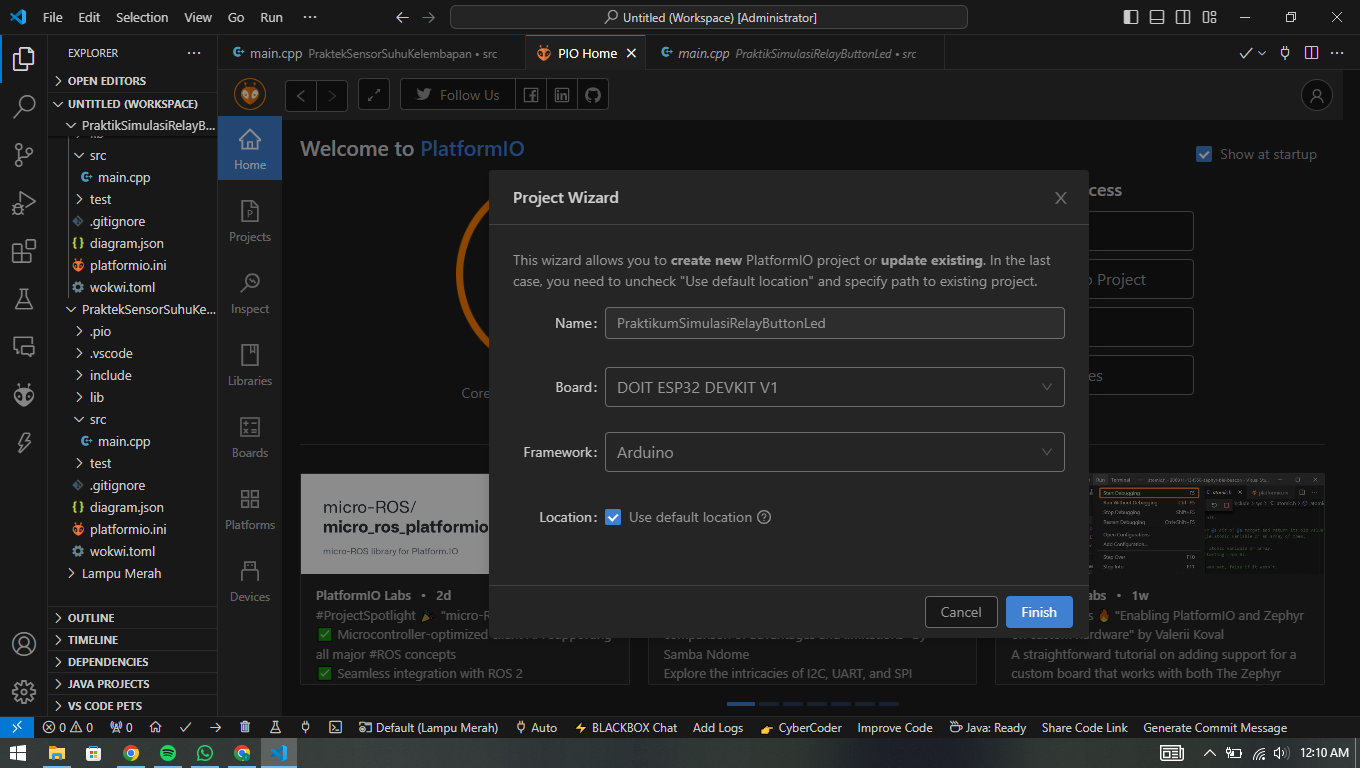
> Untuk langkah awal dalam menyusun sistem, pengkodean, dan pengujian adalah melakukan installasi ekstensi seperti **Wokwi Simulator, PlatformIO** pada V**isual Studio Code**.



> Setelah semua terinstall, klik PlatformIO lalu pilih **New Project**

****

**>** Beri nama project (SimulasiSensorJarak), kemudian pilih board menggunakan **Doit ESP32 Devkit V1**, dan framework **arduino.**

****

> Ubah syntax pada **src/main.cpp** seperti berikut, lalu compile / Build syntax.

#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);

}

> **Lalu buat file wokwi.toml** berisi syntax berikut,

[wokwi]

version = 1

firmware = '.pio\build\esp32doit-devkit-v1\firmware.bin'

elf = '.pio\build\esp32doit-devkit-v1\firmware.elf'

> Setelah itu create file **diagram.json** dan masukkan syntax berikut melalui text editor,

{

"version": 1,

"author": "Anonymous maker",

"editor": "wokwi",

"parts": [

{ "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -33.7, "left": -168.2, "attrs": {} },

{ "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -104.1, "left": -4.1, "attrs": {} }

],

"connections": [

[ "esp:TX0", "$serialMonitor:RX", "", [] ],

[ "esp:RX0", "$serialMonitor:TX", "", [] ],

[ "ultrasonic1:GND", "esp:GND.1", "black", [ "v0" ] ],

[ "ultrasonic1:VCC", "esp:VIN", "red", [ "v0" ] ],

[ "ultrasonic1:ECHO", "esp:D18", "green", [ "v0" ] ],

[ "ultrasonic1:TRIG", "esp:D5", "green", [ "v0" ] ]

],

"dependencies": {}

}

> pastikan konfigurasi yang ada di file platform,ini seperti berikut

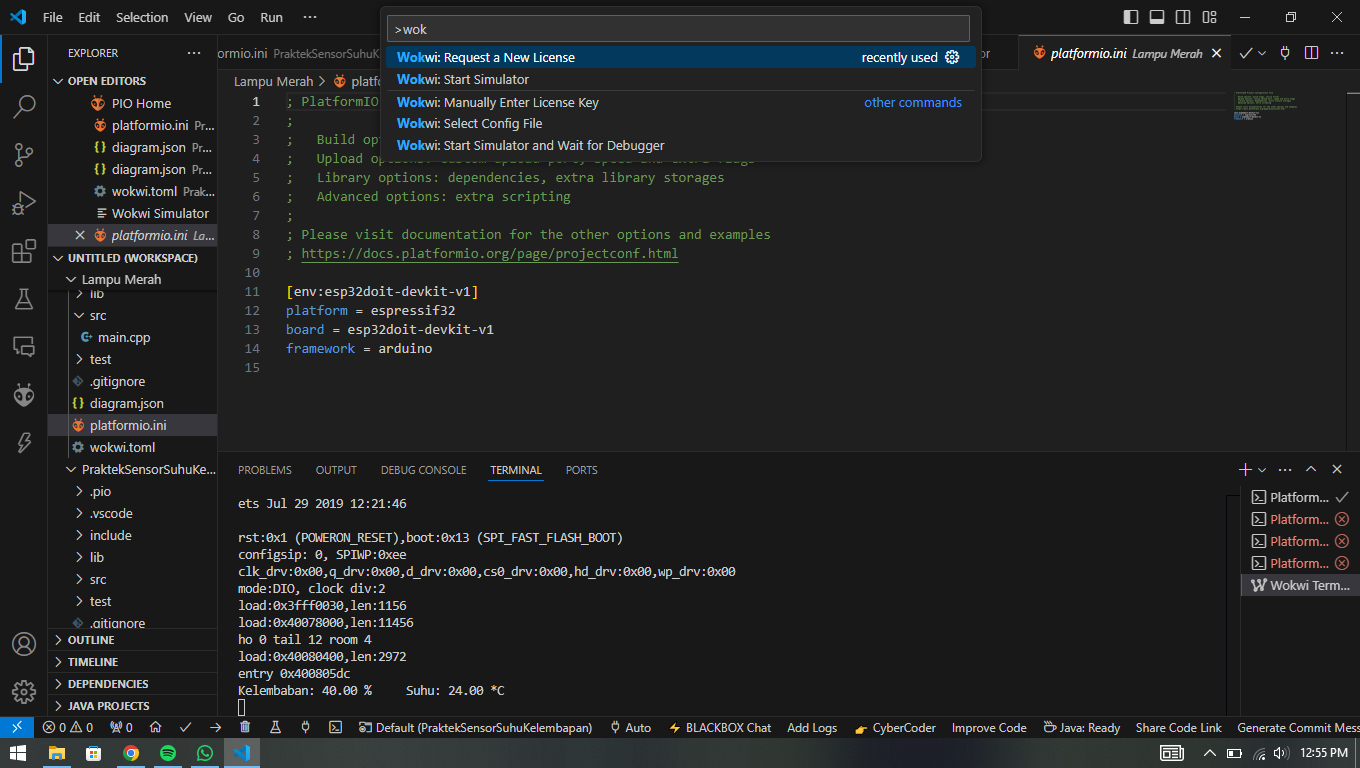
[env:esp32doit-devkit-v1]

platform = espressif32

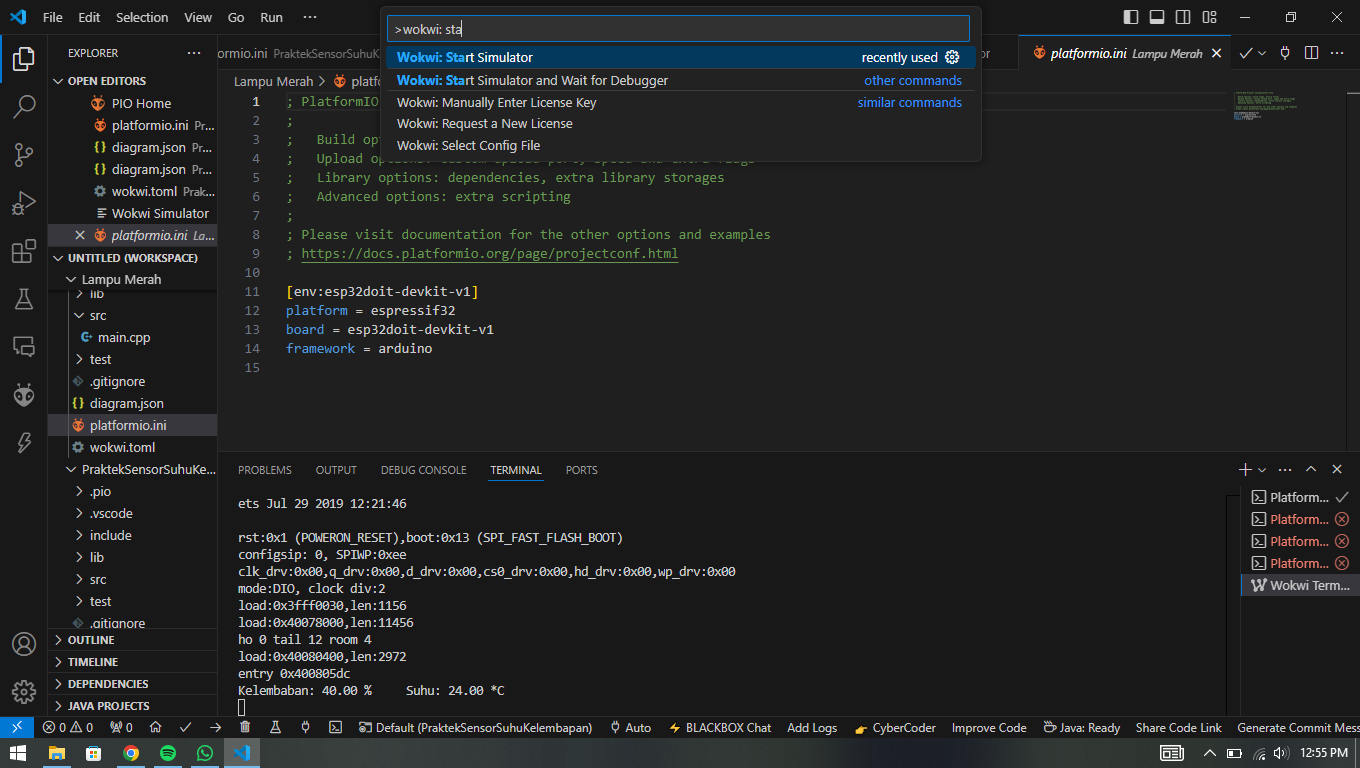
board = esp32doit-devkit-v1

framework = arduino

> Langkah selanjutnya adalah meminta lisensi wokwi untuk dapat menjalankan simulator sensornya **(Wokwi: Request a New License)**



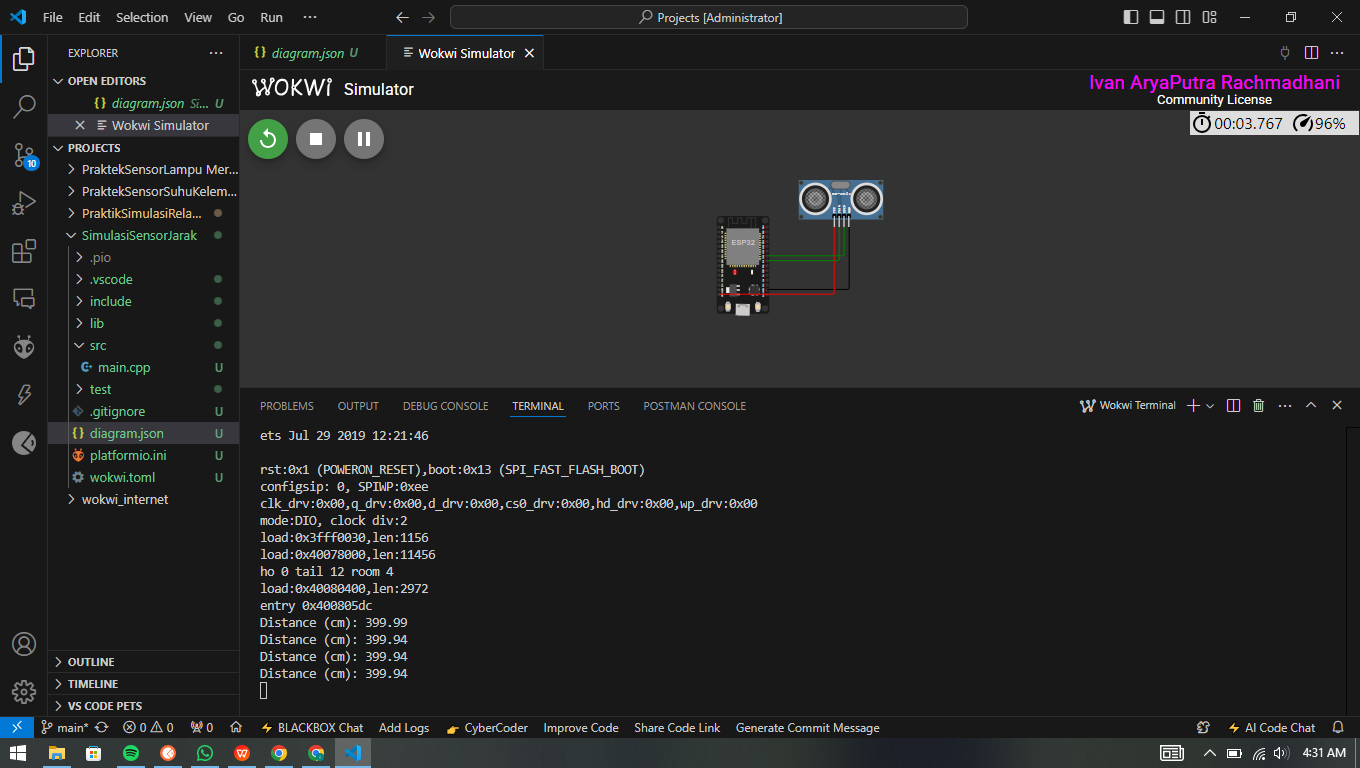
> Langkah terakhir lakukan **start simulator (Wokwi: Start Simulator)**



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

**3.1 Experimental Results (Hasil Eksperimen)**

> Data yang diperoleh :



**4. Appendix (Lampiran)**

Source Code :

#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);

}

Diagram.json

{

{

"version": 1,

"author": "Anonymous maker",

"editor": "wokwi",

"parts": [

{ "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -33.7, "left": -168.2, "attrs": {} },

{ "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -104.1, "left": -4.1, "attrs": {} }

],

"connections": [

[ "esp:TX0", "$serialMonitor:RX", "", [] ],

[ "esp:RX0", "$serialMonitor:TX", "", [] ],

[ "ultrasonic1:GND", "esp:GND.1", "black", [ "v0" ] ],

[ "ultrasonic1:VCC", "esp:VIN", "red", [ "v0" ] ],

[ "ultrasonic1:ECHO", "esp:D18", "green", [ "v0" ] ],

[ "ultrasonic1:TRIG", "esp:D5", "green", [ "v0" ] ]

],

"dependencies": {}

}