LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

# **Praktik Simulasi ESP32 & Sensor Suhu Kelembaban**



*Dina Hanifa*

*Fakultas Vokasi, Universitas Brawijaya*

*Email :* [*dinahanifa@student.ub.ac.id*](mailto:dinahanifa@student.ub.ac.id)

**Abstrak**

Eksperimen ini bertujuan untuk merancang dan mengimplementasikan sistem sensor suhu dan kelembaban berbasis mikrokontroler. Proses mencakup perakitan rangkaian, pemrograman mikrokontroler menggunakan Visual Studio Code (VS Code), dan pengujian kinerja sistem. Hasilnya menunjukkan bahwa sensor tersebut dapat mendeteksi suhu kelembaban ruangan. Kesimpulan dari eksperimen ini adalah bahwa rangkaian dapat dikendalikan dengan baik melalui kode pemrograman yang sesuai.

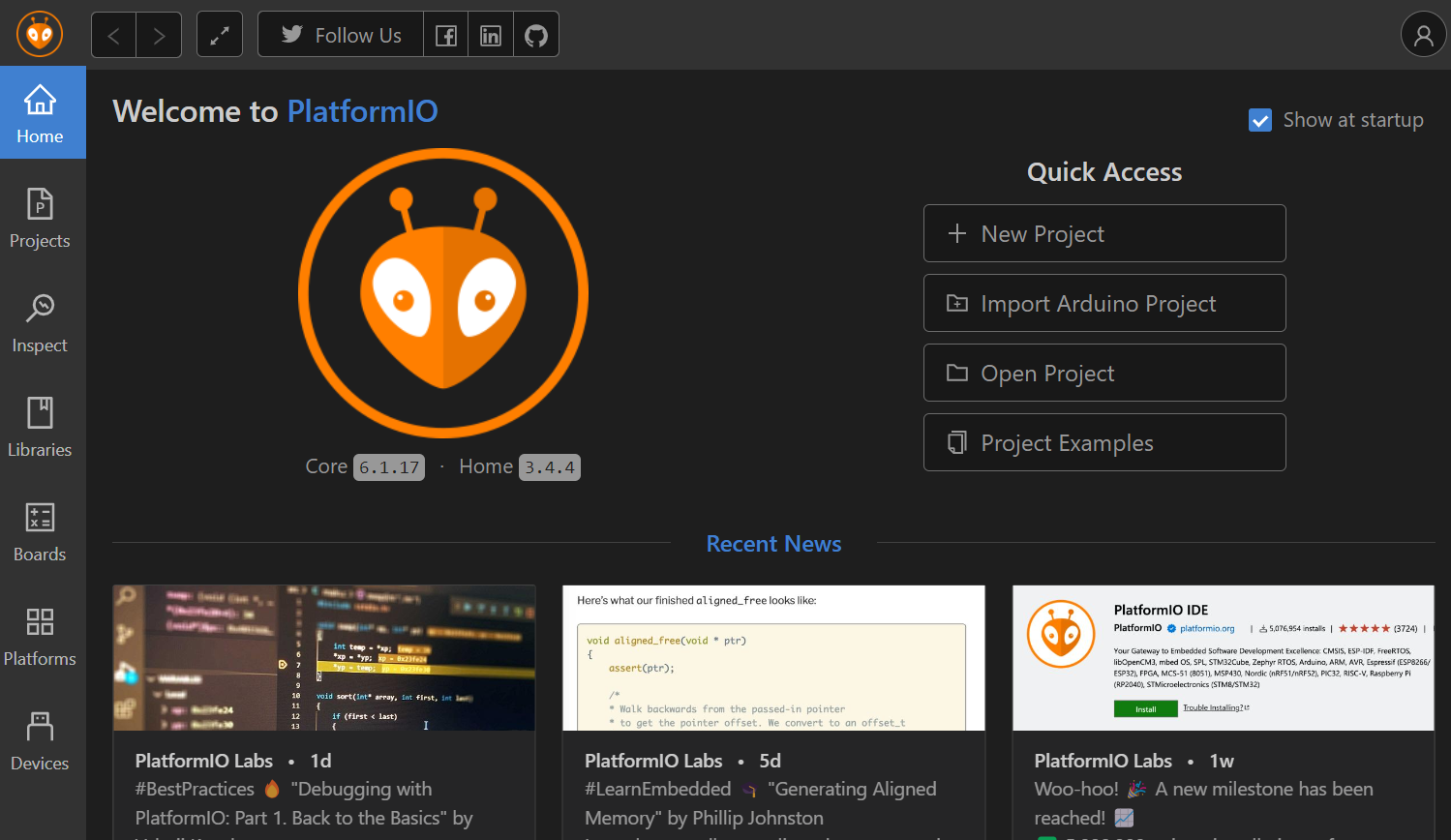
*Kata kunci: Suhu, IoT, Mikrokontroler, Pemrograman, Sensor*

1. **Pendahuluan** 
   1. **Latar Belakang**

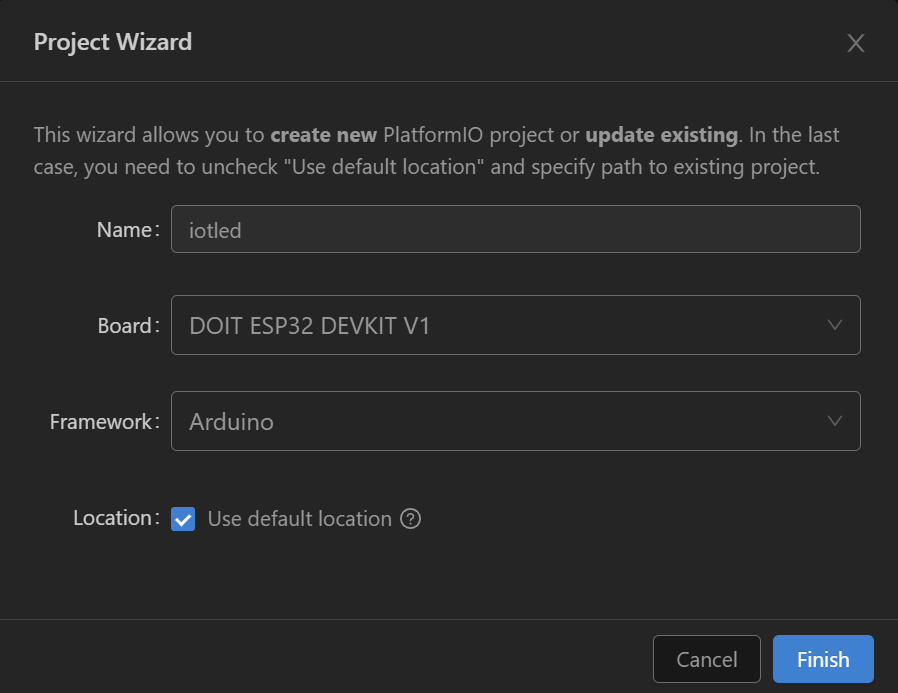
Saat ini permintaan terhadap otomatisasi dan sistem intelijen sangat tinggi. Itu sebabnya masyarakat menunjukkan ketertarikan terhadap perangkat pintar. Contohnya masyarakat dapat mengontrol atau memonitor alat-alat rumah tangga mereka melalui web atau aplikasi melalui telepon genggam. Internet of Things (IoT) yang dapat membuat alat-alat atau perangkat keras tersebut dapat berkomunikasi, bertukar data, dan saling mengendalikan melalui web atau aplikasi smartphone. Suhu dan kelembaban udara di lingkungan pun dapat dimonitor melalui web dan smartphone menggunakan (IoT) agar udara di lingkungan tersebut tetap sehat dan terjaga.

* 1. **Tujuan Eksperimen**

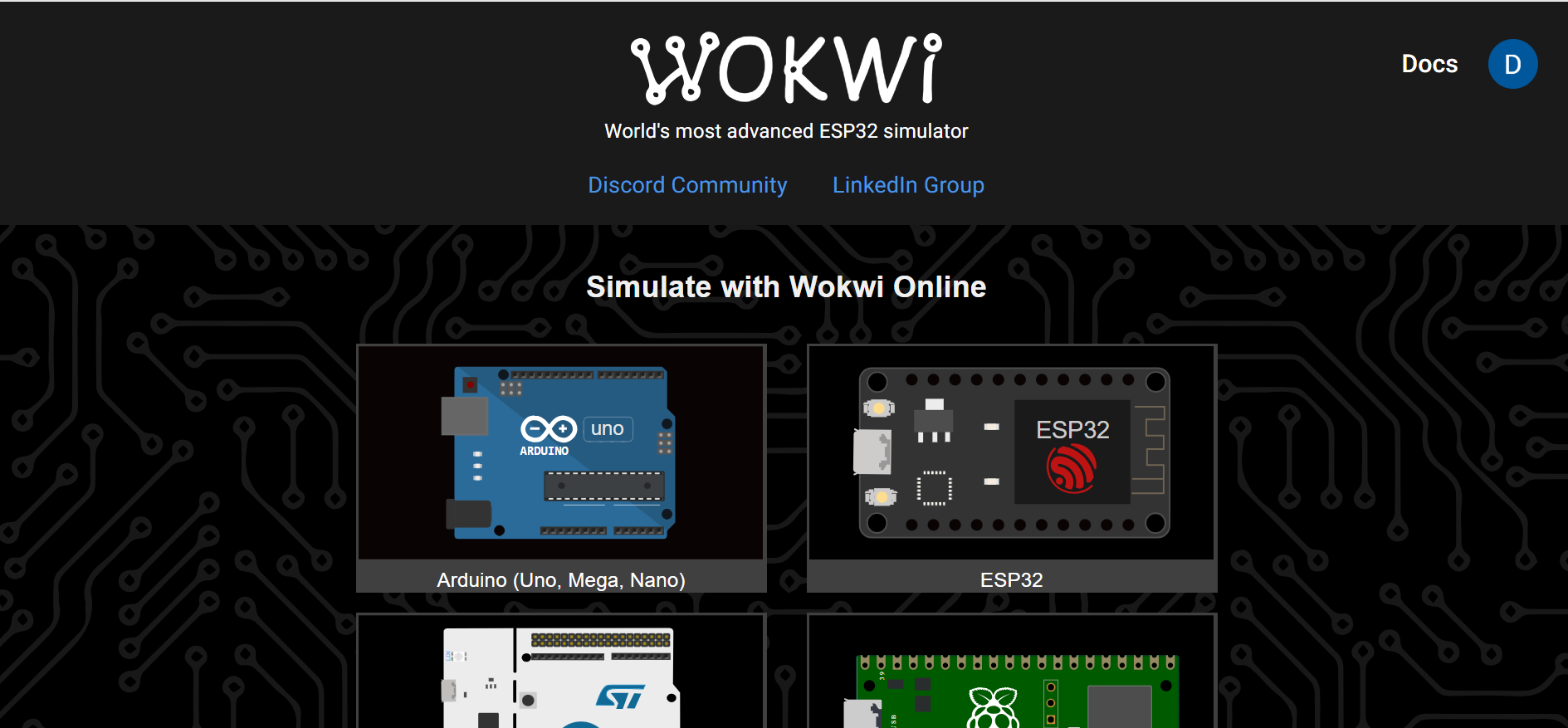
1. Mempelajari sistem pendeteksi suhu kelembaban ruangan.
2. Merancang dan mengimplementasikan sensor kelembaban suhu menggunakan mikrokontroler.
3. Menguji kinerja sistem dan menganalisis hasilnya.
4. **Metodologi**
   1. **Alat dan Bahan**
5. Perangkat komputer dengan koneksi internet
6. Mikrokontroler ESP32
7. Sensor DHT22
8. Software Visual Studio Code (VS Code)
9. Akses ke situs Wokwi (<https://wokwi.com>)
   1. **Langkah Implementasi**
10. Buka PlatformIO di VSCode lalu buat project baru



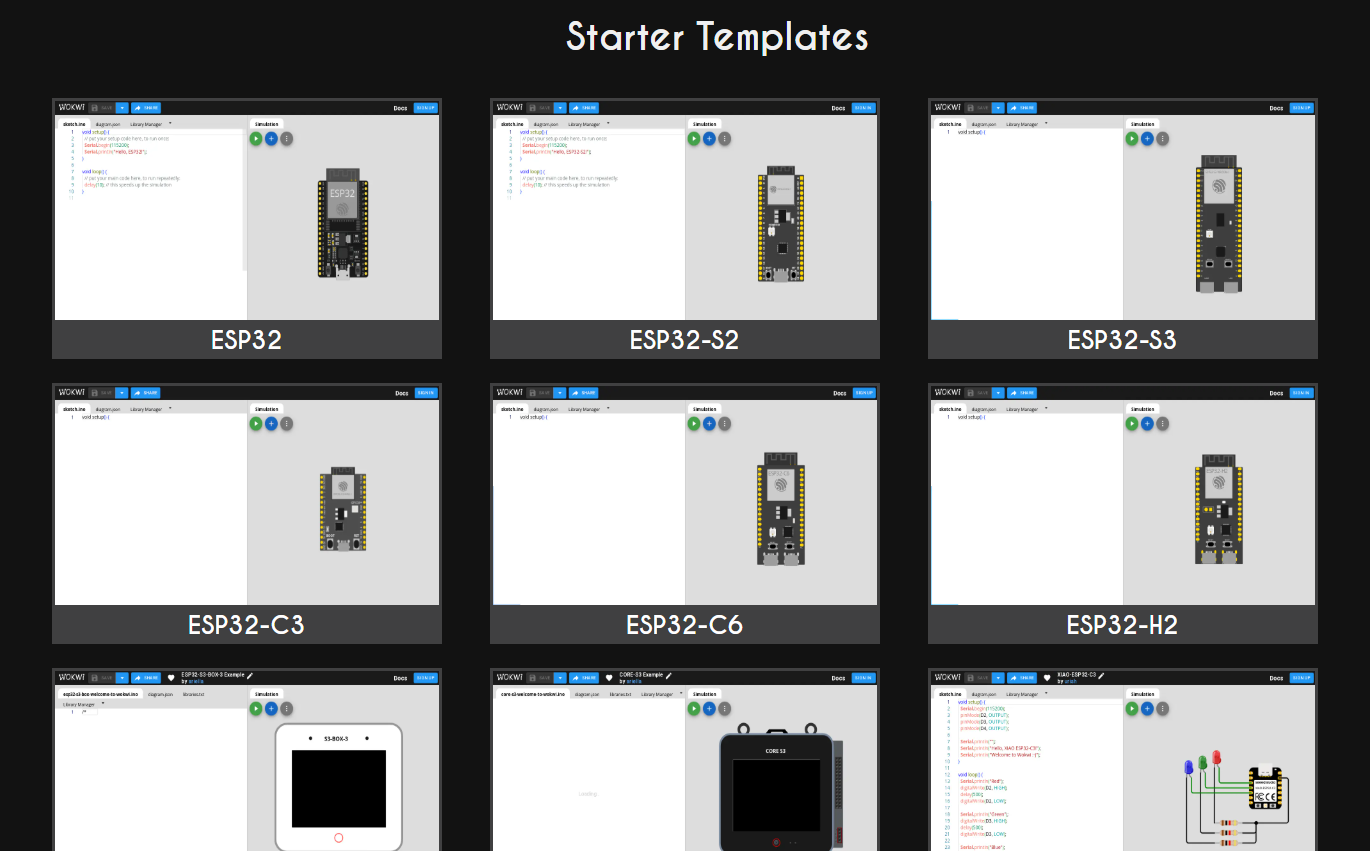
Sesuaikan nama project yang ingin dibuat, pilih Board DOIT ESP32 DEVKIT V1 dan gunakan framework Arduino, setelah itu tekan finish



1. Buka situs wokwi dan pilih mikrokontroler ESP32

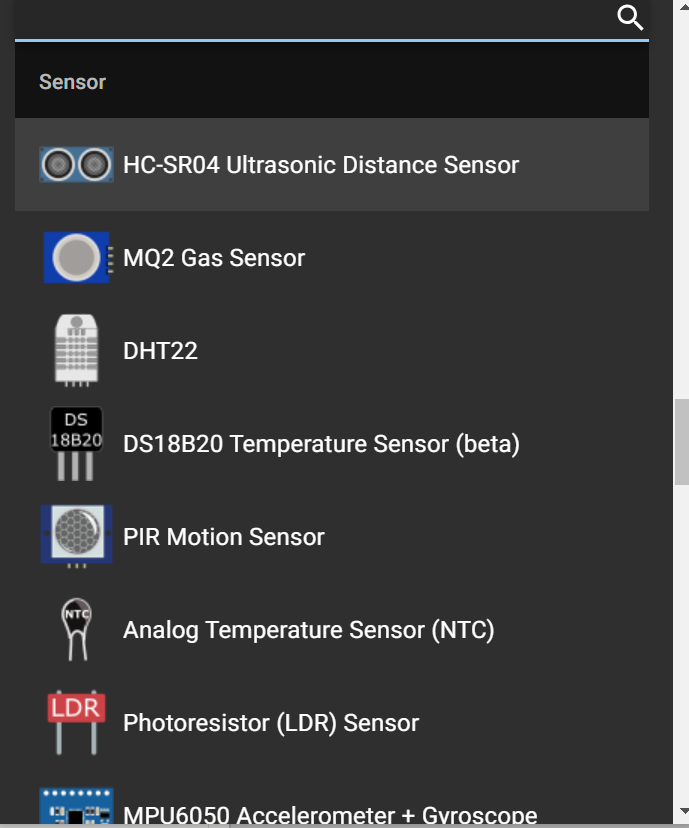


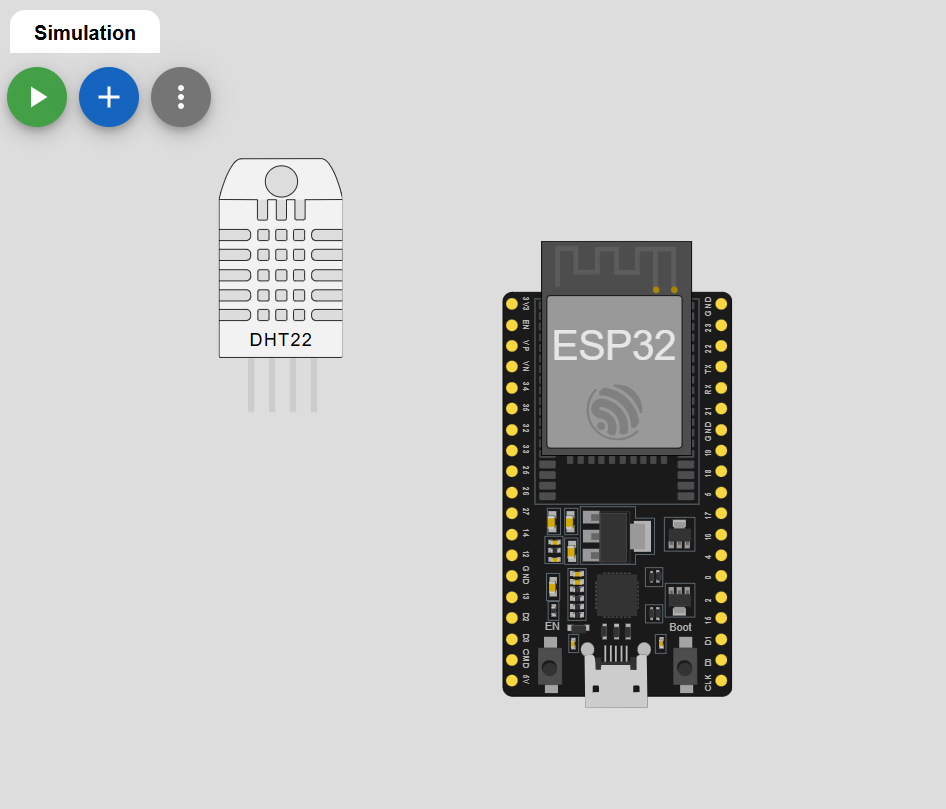
Pilih ESP32 di Starter Templates, maka tampilan akan seperti ini



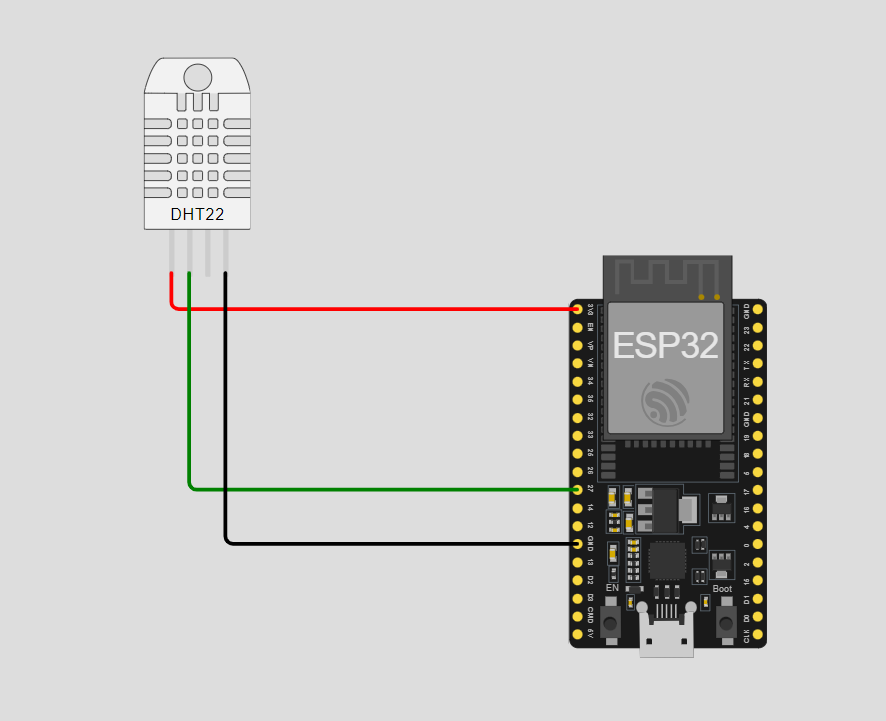
1. Klik icon tambah lalu scroll pada bagian sensor dan pilih DHT22







1. Menghubungkan kabel DHT22 ke ESP32

****

Pin VCC (DHT22) kabel warna merah ke 3.3V (ESP32)

Pin GND (DHT22) kabel warna hitam ke GND (ESP32)

Pin Data (DHT22) kabel warna hijau ke GPIO (ESP32

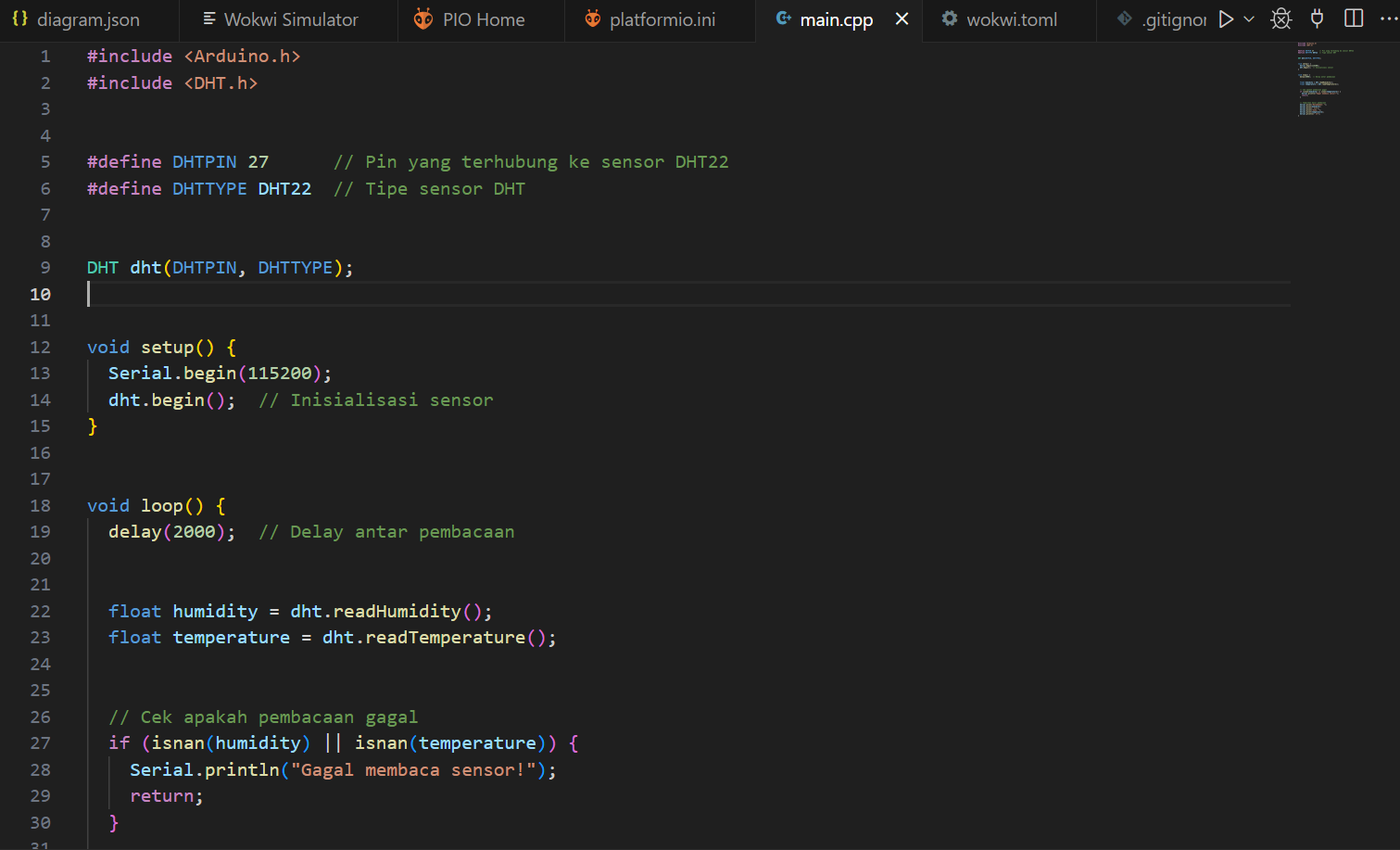
1. Tambahkan kode relay sebagai output untuk arduino, kemudian disusul baris kedua sebagai coman void setup, lalu void loop sebagai codingan untuk langkah kerja dan diakhiri dengan } sebagai penutup dalam perintah codingan/pemograman seperti pada uraian berikut ini: .







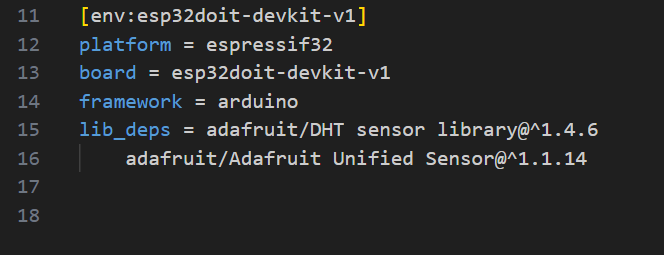
1. Tulis koding C++ untuk project ini pada file **src/main.cpp** seperti dibawah ini

****

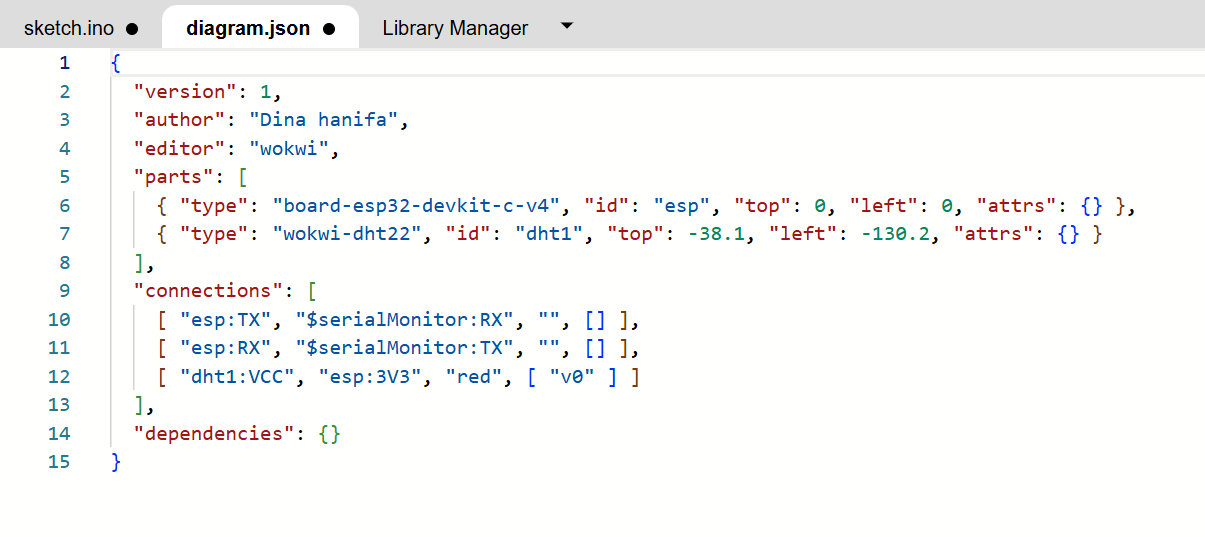
****

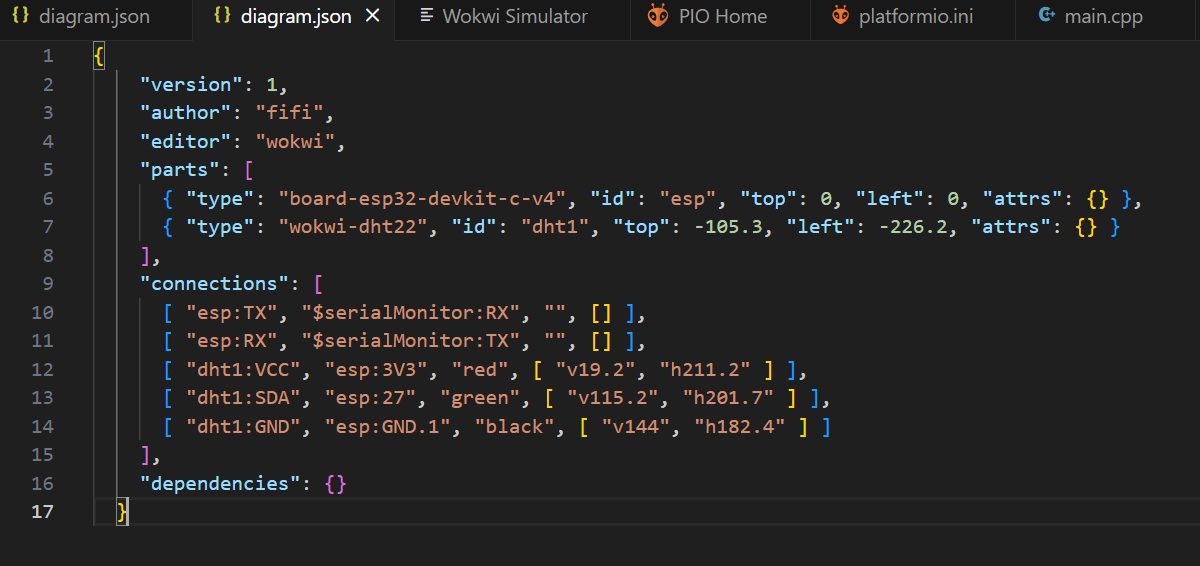
Program ini menggunakan ESP32 untuk membaca suhu dan kelembaban dari sensor DHT22 setiap 2 detik, menampilkan hasil di Serial Monitor, serta menangani kesalahan jika pembacaan gagal.

1. Mengisi kode pada platform.ini

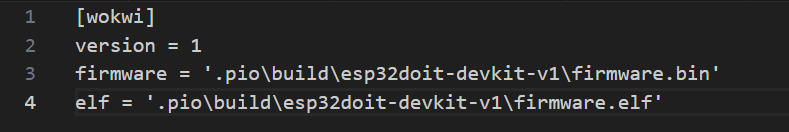


1. Buat file baru diagram.json dan copy paste dari diagram.json pada platform online wokwi.com

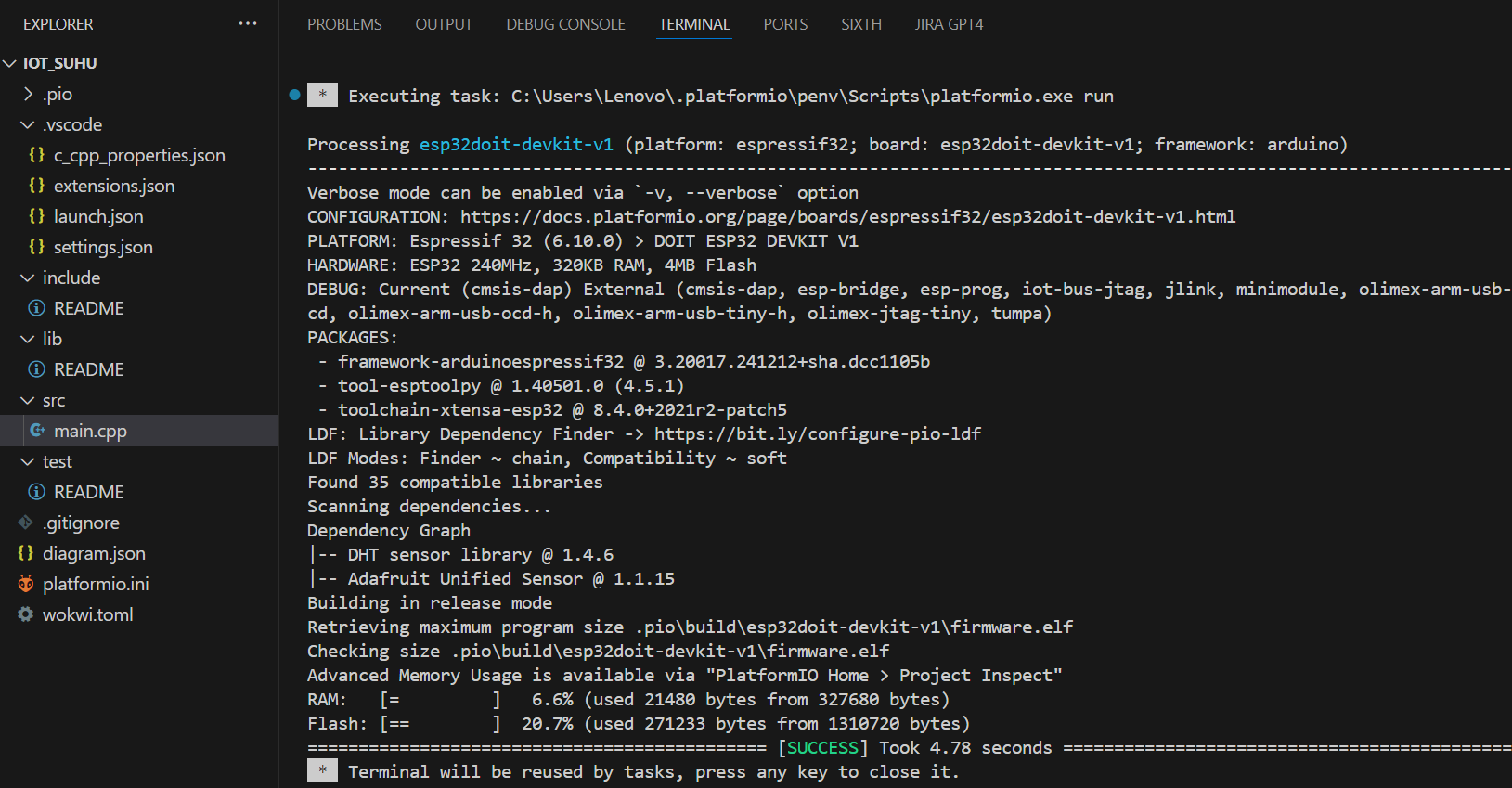




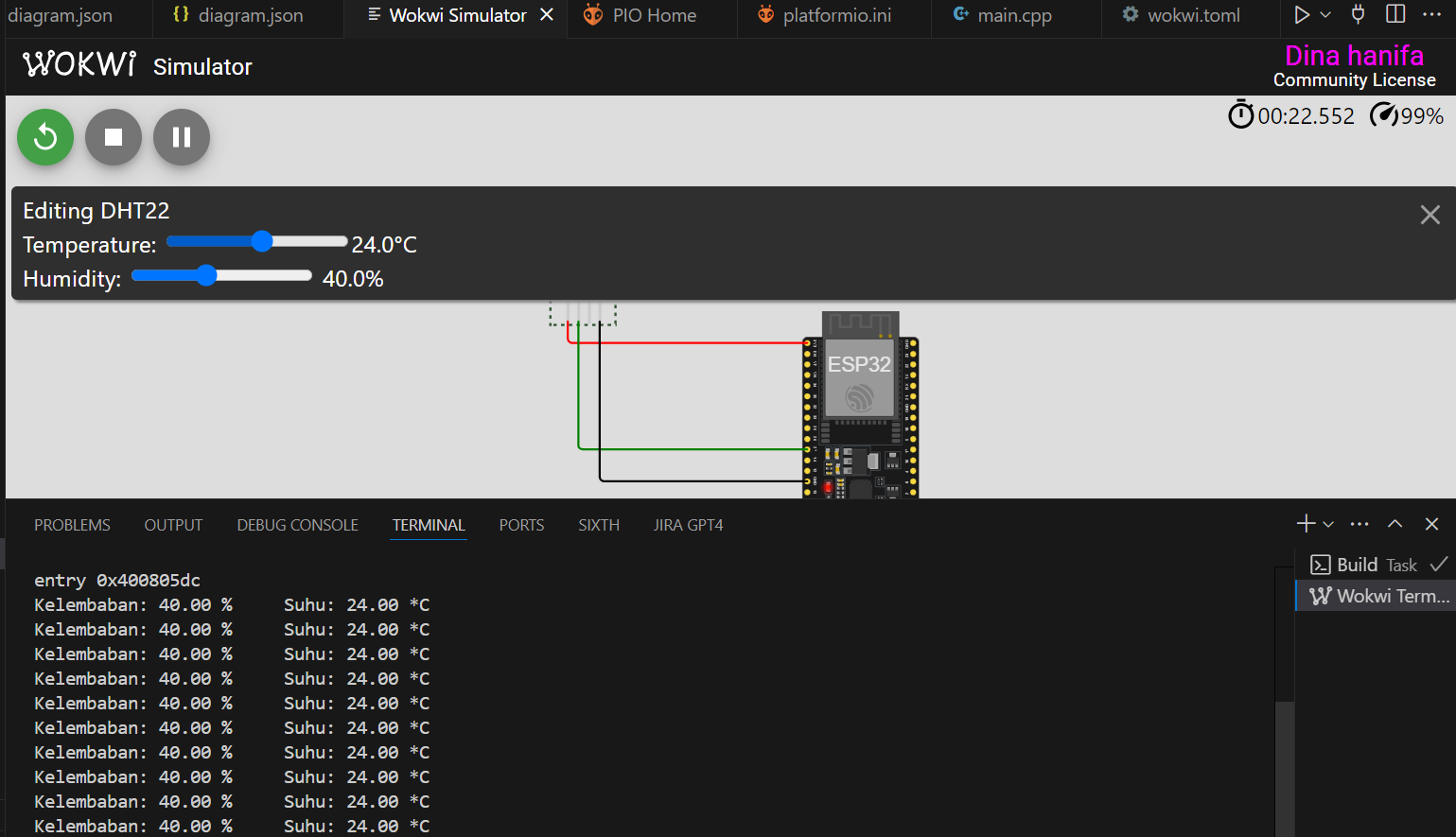
1. Buat file baru **wokwi.toml**, dan isikan file tersebut dengan koding sebagai berikut :



1. Lakukan compile pada file main.cpp

****

1. Jalankan simulasi



1. **Hasil dan Pembahasan** 
   1. **Hasil Eksperimen**

* Setelah kode diunggah, ESP32 berhasil membaca data dari sensor dan menampilkannya melalui Serial Monitor setiap 2 detik. Pada hasil output yang diperoleh yaitu kelembaban 40.00 %, suhu: 24.00 \*C
* Hasil ini menunjukkan bahwa sensor DHT22 mampu membaca suhu dan kelembaban dengan nilai yang stabil. Selain itu, jika sensor tidak terbaca dengan benar, program akan menampilkan pesan error **"Gagal membaca sensor!"**, yang membantu dalam proses debugging.

