

LAPORAN PRAKTIKUM INTERNET OF THINGS (IOT)
IMPLEMENTASI MODEL TINYML UNTUK KLASIFIKASI
DATASET IRIS MENGGUNAKAN ESP32

FAKULTAS VOKASI
UNIVERSITAS BRAWIJAYA

Nasywa Anindya Q.E

Fakultas Vokasi Universitas Brawijaya

Email : nasywaanindya@student.ub.ac.id

ABSTRAK

Proyek ini bertujuan menerapkan konsep Tiny Machine Learning (TinyML) pada mikrokontroler ESP32. Model klasifikasi IRIS yang telah dilatih sebelumnya diintegrasikan ke dalam ESP32 menggunakan TensorFlow Lite dan pustaka EloquentTinyML, sehingga perangkat mampu memprediksi jenis bunga berdasarkan panjang dan lebar sepal maupun petal. Seluruh proses prediksi dijalankan secara lokal tanpa memerlukan koneksi internet setelah pemrograman selesai. Namun, koneksi tetap dibutuhkan selama tahap pengembangan dan simulasi. Simulasi proyek dilakukan pada platform Wokwi, dengan hasil pemantauan melalui Serial Monitor di Visual Studio Code.

Kata Kunci: ESP32, Wokwi, TinyML

ABSTRACT

This project aims to apply the Tiny Machine Learning (TinyML) concept on an ESP32 microcontroller. A pre-trained IRIS classification model is embedded into the ESP32 using TensorFlow Lite and the EloquentTinyML library, enabling the device to predict flower species based on sepal and petal length and width. The prediction runs locally without the need for an internet connection once programmed. However, a connection is necessary during development and simulation phases. The project simulation is performed on the Wokwi platform, and the result are observed using the Serial Monitor in Visual Studio Code.

Keywords: ESP32, Wokwi, TinyML

1. PENDAHULUAN

1.1. Latar Belakang

Perkembangan pesat teknologi kecerdasan buatan (AI) dan machine learning dalam beberapa tahun terakhir telah membuka peluang baru, salah satunya melalui Tiny Machine Learning (TinyML). TinyML memungkinkan model pembelajaran mesin berjalan pada perangkat kecil dengan sumber daya terbatas, seperti mikrokontroler, tanpa bergantung pada koneksi internet atau layanan cloud. Hal ini sangat relevan untuk aplikasi Internet of Things (IoT) yang membutuhkan respon cepat, hemat daya, dan keamanan data. ESP32 menjadi pilihan tepat untuk mendukung pengembangan TinyML karena memiliki kemampuan pemrosesan yang cukup baik dan fitur konektivitas lengkap. Dengan dukungan TensorFlow Lite yang telah dioptimalkan, ESP32 mampu menjalankan model machine learning yang biasanya hanya digunakan di perangkat dengan sumber daya lebih besar.

Sebagai studi kasus, digunakan dataset IRIS — dataset klasik dalam dunia machine learning yang sering dipakai untuk pengenalan pola. Dataset ini memiliki fitur sederhana berupa panjang dan lebar sepal maupun petal yang dikategorikan ke dalam tiga spesies bunga iris. Proyek ini menunjukkan bahwa ESP32 mampu menjalankan tugas AI dengan baik menggunakan model klasifikasi IRIS. Meskipun TinyML mendukung inferensi secara offline, tahap pengembangan model, konversi ke format TensorFlow Lite, dan simulasi menggunakan platform seperti Wokwi tetap memerlukan koneksi internet. Hal ini menjadi aspek penting yang perlu diperhatikan dalam penerapan TinyML pada skala lebih luas.

Proyek ini bertujuan untuk memperdalam pemahaman di bidang AI embedded dan IoT serta mengembangkan sistem cerdas yang mandiri dengan menggabungkan teknologi machine learning pada perangkat tertanam.

1.2. Tujuan

Tujuan praktikum ini meliputi:

1. Mengintegrasikan model klasifikasi IRIS berbasis TensorFlow Lite ke mikrokontroler ESP32.
2. Memahami alur TinyML mulai dari pelatihan model hingga implementasi.
3. Melakukan simulasi dan pengujian model pada platform Wokwi dengan bantuan VSCode.

2. METODOLOGI

2.1. Alat dan Bahan

2.1.1. Alat

- Laptop
- Wokwi Simulator
- Visual Studio Code (VSCode)

2.1.2. Bahan

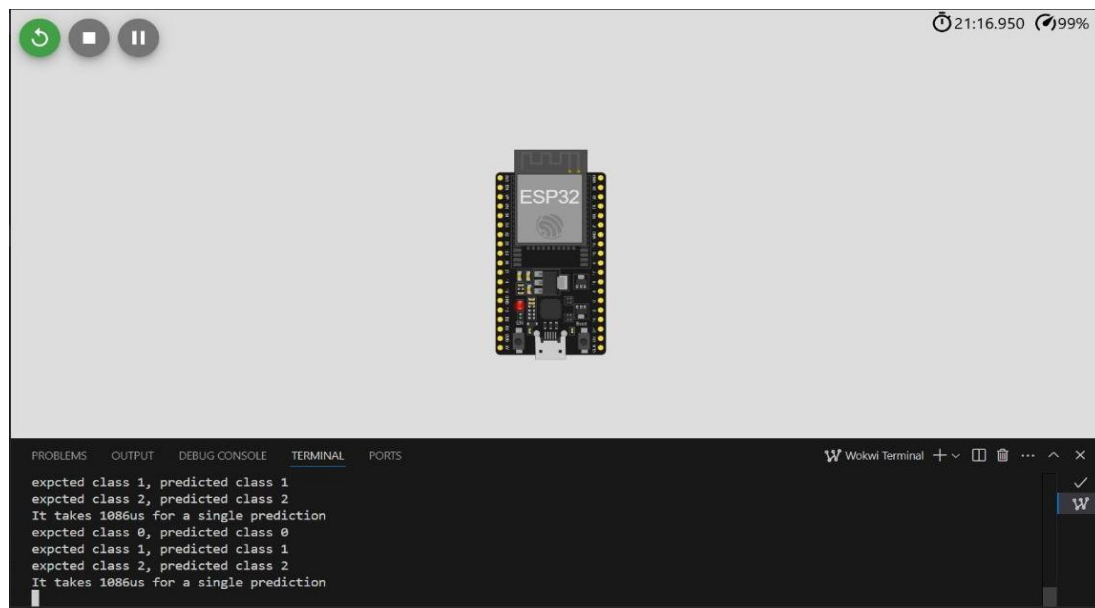
- ESP32 (simulasi melalui Wokwi)
- Bahasa pemrograman C++ dengan Pustaka Arduino
- Dataset IRIS
- Library:
 - a. Eloquentarduino/tflm_esp32
 - b. EloqueTinyML

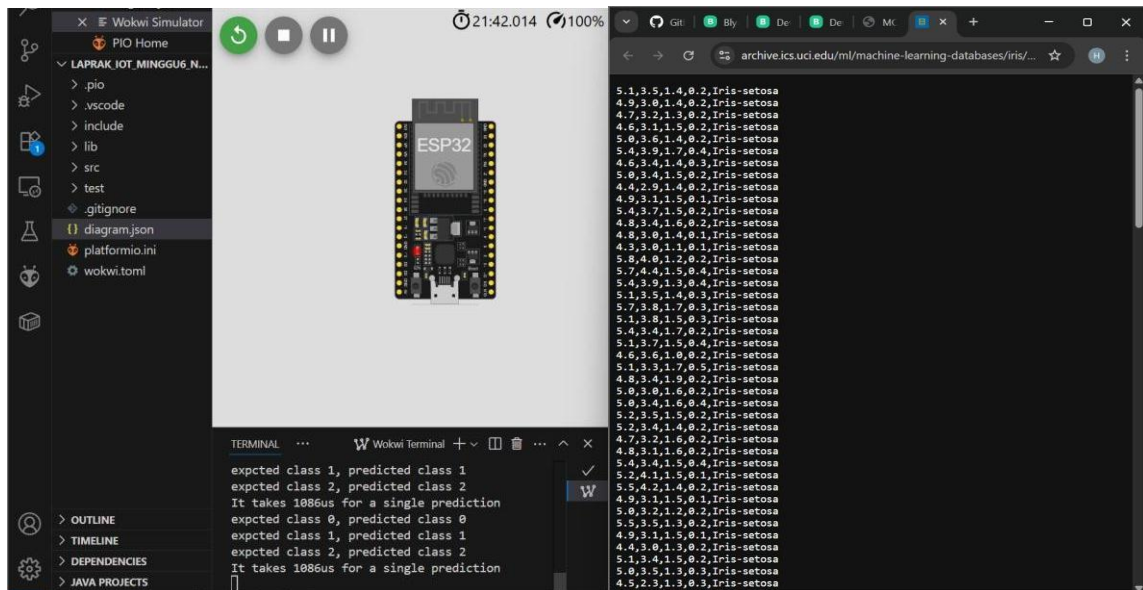
3. LANGKAH IMPLEMENTASI

- Jalankan Visual Studio Code.
- Pasang ekstensi PlatformIO.
- Buat proyek baru menggunakan board DOIT ESP32 DEVKIT V1.
- Tambahkan library tflm_esp32 dan EloqueTinyML.
- Salin file iris_model.h hasil konversi model TensorFlow Lite ke folder *include*.
- Buat file program main.cpp di folder *src* untuk menulis kode utama.
- Jalankan program.
- Periksa hasil prediksi kelas IRIS melalui Serial Monitor.

4. HASIL DAN PEMBAHASAN

4.1. Hasil Eksperimen





4.2. Pembahasan

Dengan waktu inferensi sekitar 1 milidetik, implementasi TinyML pada ESP32 melalui simulasi di Wokwi membuktikan bahwa mikrokontroler berukuran kecil mampu menjalankan model AI secara efektif. Hal ini menegaskan bahwa perangkat edge dapat melaksanakan tugas AI secara real-time dengan konsumsi daya yang sangat rendah. Model machine learning yang dimuat berjalan sepenuhnya secara lokal di dalam perangkat, sehingga proses inferensi tidak memerlukan koneksi internet. Namun, perlu dicatat bahwa koneksi internet tetap dibutuhkan pada tahap pengembangan dan simulasi. Kondisi ini memperlihatkan bahwa TinyML sangat mendukung aplikasi yang memerlukan tingkat kemandirian tinggi, seperti pada perangkat IoT yang tersebar luas.

Dataset Iris digunakan sebagai data pelatihan awal yang disajikan dalam bentuk tabel berisi panjang dan lebar sepal serta petal dari tiga spesies Iris. Data tersebut digunakan untuk melatih model machine learning agar dapat mengenali pola dan membuat klasifikasi. Hasil dari proses pelatihan ini adalah model IRIS yang telah dilatih (trained model) dan dikompilasi dalam bentuk array biner const unsigned char irisModel[]. Model ini sudah melalui proses pembelajaran untuk menyesuaikan parameter internal seperti bobot dan bias pada arsitektur jaringan sarafnya sehingga mampu mengklasifikasikan spesies Iris berdasarkan input yang diberikan. Dengan demikian, Iris Dataset berperan sebagai sumber data untuk proses belajar, sedangkan Iris Model merupakan hasil konkret dari proses pembelajaran tersebut yang siap digunakan untuk melakukan prediksi terhadap data baru, seperti nilai-nilai x_0 , x_1 , dan x_2 pada aplikasi nyata.

5. LAMPIRAN

a. Kode Program Main.cpp

```
#include <Arduino.h>

/**
 * Run a TensorFlow model to predict the IRIS dataset
 * For a complete guide, visit
 * https://eloquentarduino.com/tensorflow-lite-esp32
 */
```

```

// replace with your own model
// include BEFORE <eloquent_tinyml.h>!
#include "iris_model.h"
// include the runtime specific for your board
// either tflm_esp32 or tflm_cortexm
#include <tflm_esp32.h>
// now you can include the eloquent tinyml wrapper
#include <eloquent_tinyml.h>

// this is trial-and-error process
// when developing a new model, start with a high value
// (e.g. 10000), then decrease until the model stops
// working as expected
#define ARENA_SIZE 2000

Eloquent::TF::Sequential<TF_NUM_OPS, ARENA_SIZE> tf;
//Eloquent::TinyML::TfLite<4,3,ARENA_SIZE> tf;

/**
 *
 */
void setup() {
    Serial.begin(115200);
    delay(3000);
    Serial.println("__TENSORFLOW IRIS__");

    // configure input/output
    // (not mandatory if you generated the .h model
    // using the everywhereml Python package)
    tf.setNumInputs(4);
    tf.setNumOutputs(3);
    // add required ops
    // (not mandatory if you generated the .h model
    // using the everywhereml Python package)
    tf.resolver.AddFullyConnected();
    tf.resolver.AddSoftmax();

    while (!tf.begin(irisModel).isOk())
        Serial.println(tf.exception.toString());
}

```

```
}
```

```
void loop() {  
    // x0, x1, x2 are defined in the irisModel.h file  
    //  
https://github.com/eloquentarduino/EloquentTinyML/tree/main/examples/IrisExample/irisModel.h  
    h  
    if (!tf.predict(x0).isOk()) {  
        Serial.println(tf.exception.toString());  
        return;  
    }  
  
    Serial.print("expcted class 0, predicted class ");  
    Serial.println(tf.classification);  
  
    // classify sample from class 1  
    if (!tf.predict(x1).isOk()) {  
        Serial.println(tf.exception.toString());  
        return;  
    }  
  
    Serial.print("expcted class 1, predicted class ");  
    Serial.println(tf.classification);  
  
    // classify sample from class 2  
    if (!tf.predict(x2).isOk()) {  
        Serial.println(tf.exception.toString());  
        return;  
    }  
  
    Serial.print("expcted class 2, predicted class ");  
    Serial.println(tf.classification);  
  
    // how long does it take to run a single prediction?  
    Serial.print("It takes ");  
    Serial.print(tf.benchmark.microseconds());  
    Serial.println("us for a single prediction");  
}
```

```
    delay(1000);  
  }
```

b. Kode Program diagram.json

```
{  
  "version": 1,  
  "author": "Nasywa Anindya",  
  "editor": "wokwi",  
  "parts": [ { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 0, "left": 0, "attrs": { } } ],  
  "connections": [ [ "esp:TX", "$serialMonitor:RX", "", [] ], [ "esp:RX", "$serialMonitor:TX", "",  
    [] ] ],  
}
```

c. Kode iris_model.h

5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
5.4,3.7,1.5,0.2,Iris-setosa
4.8,3.4,1.6,0.2,Iris-setosa
4.8,3.0,1.4,0.1,Iris-setosa
4.3,3.0,1.1,0.1,Iris-setosa
5.8,4.0,1.2,0.2,Iris-setosa
5.7,4.4,1.5,0.4,Iris-setosa
5.4,3.9,1.3,0.4,Iris-setosa
5.1,3.5,1.4,0.3,Iris-setosa
5.7,3.8,1.7,0.3,Iris-setosa
5.1,3.8,1.5,0.3,Iris-setosa
5.4,3.4,1.7,0.2,Iris-setosa
5.1,3.7,1.5,0.4,Iris-setosa
4.6,3.6,1.0,0.2,Iris-setosa
5.1,3.3,1.7,0.5,Iris-setosa
4.8,3.4,1.9,0.2,Iris-setosa
5.0,3.0,1.6,0.2,Iris-setosa
5.0,3.4,1.6,0.4,Iris-setosa

5.2,3.5,1.5,0.2,Iris-setosa
5.2,3.4,1.4,0.2,Iris-setosa
4.7,3.2,1.6,0.2,Iris-setosa
4.8,3.1,1.6,0.2,Iris-setosa
5.4,3.4,1.5,0.4,Iris-setosa
5.2,4.1,1.5,0.1,Iris-setosa
5.5,4.2,1.4,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
5.0,3.2,1.2,0.2,Iris-setosa
5.5,3.5,1.3,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
4.4,3.0,1.3,0.2,Iris-setosa
5.1,3.4,1.5,0.2,Iris-setosa
5.0,3.5,1.3,0.3,Iris-setosa
4.5,2.3,1.3,0.3,Iris-setosa
4.4,3.2,1.3,0.2,Iris-setosa
5.0,3.5,1.6,0.6,Iris-setosa
5.1,3.8,1.9,0.4,Iris-setosa
4.8,3.0,1.4,0.3,Iris-setosa
5.1,3.8,1.6,0.2,Iris-setosa
4.6,3.2,1.4,0.2,Iris-setosa
5.3,3.7,1.5,0.2,Iris-setosa
5.0,3.3,1.4,0.2,Iris-setosa
7.0,3.2,4.7,1.4,Iris-versicolor
6.4,3.2,4.5,1.5,Iris-versicolor
6.9,3.1,4.9,1.5,Iris-versicolor
5.5,2.3,4.0,1.3,Iris-versicolor
6.5,2.8,4.6,1.5,Iris-versicolor
5.7,2.8,4.5,1.3,Iris-versicolor
6.3,3.3,4.7,1.6,Iris-versicolor
4.9,2.4,3.3,1.0,Iris-versicolor
6.6,2.9,4.6,1.3,Iris-versicolor
5.2,2.7,3.9,1.4,Iris-versicolor
5.0,2.0,3.5,1.0,Iris-versicolor
5.9,3.0,4.2,1.5,Iris-versicolor
6.0,2.2,4.0,1.0,Iris-versicolor
6.1,2.9,4.7,1.4,Iris-versicolor
5.6,2.9,3.6,1.3,Iris-versicolor
6.7,3.1,4.4,1.4,Iris-versicolor

5.6,3.0,4.5,1.5,Iris-versicolor
5.8,2.7,4.1,1.0,Iris-versicolor
6.2,2.2,4.5,1.5,Iris-versicolor
5.6,2.5,3.9,1.1,Iris-versicolor
5.9,3.2,4.8,1.8,Iris-versicolor
6.1,2.8,4.0,1.3,Iris-versicolor
6.3,2.5,4.9,1.5,Iris-versicolor
6.1,2.8,4.7,1.2,Iris-versicolor
6.4,2.9,4.3,1.3,Iris-versicolor
6.6,3.0,4.4,1.4,Iris-versicolor
6.8,2.8,4.8,1.4,Iris-versicolor
6.7,3.0,5.0,1.7,Iris-versicolor
6.0,2.9,4.5,1.5,Iris-versicolor
5.7,2.6,3.5,1.0,Iris-versicolor
5.5,2.4,3.8,1.1,Iris-versicolor
5.5,2.4,3.7,1.0,Iris-versicolor
5.8,2.7,3.9,1.2,Iris-versicolor
6.0,2.7,5.1,1.6,Iris-versicolor
5.4,3.0,4.5,1.5,Iris-versicolor
6.0,3.4,4.5,1.6,Iris-versicolor
6.7,3.1,4.7,1.5,Iris-versicolor
6.3,2.3,4.4,1.3,Iris-versicolor
5.6,3.0,4.1,1.3,Iris-versicolor
5.5,2.5,4.0,1.3,Iris-versicolor
5.5,2.6,4.4,1.2,Iris-versicolor
6.1,3.0,4.6,1.4,Iris-versicolor
5.8,2.6,4.0,1.2,Iris-versicolor
5.0,2.3,3.3,1.0,Iris-versicolor
5.6,2.7,4.2,1.3,Iris-versicolor
5.7,3.0,4.2,1.2,Iris-versicolor
5.7,2.9,4.2,1.3,Iris-versicolor
6.2,2.9,4.3,1.3,Iris-versicolor
5.1,2.5,3.0,1.1,Iris-versicolor
5.7,2.8,4.1,1.3,Iris-versicolor
6.3,3.3,6.0,2.5,Iris-virginica
5.8,2.7,5.1,1.9,Iris-virginica
7.1,3.0,5.9,2.1,Iris-virginica
6.3,2.9,5.6,1.8,Iris-virginica
6.5,3.0,5.8,2.2,Iris-virginica

7.6,3.0,6.6,2.1,Iris-virginica
4.9,2.5,4.5,1.7,Iris-virginica
7.3,2.9,6.3,1.8,Iris-virginica
6.7,2.5,5.8,1.8,Iris-virginica
7.2,3.6,6.1,2.5,Iris-virginica
6.5,3.2,5.1,2.0,Iris-virginica
6.4,2.7,5.3,1.9,Iris-virginica
6.8,3.0,5.5,2.1,Iris-virginica
5.7,2.5,5.0,2.0,Iris-virginica
5.8,2.8,5.1,2.4,Iris-virginica
6.4,3.2,5.3,2.3,Iris-virginica
6.5,3.0,5.5,1.8,Iris-virginica
7.7,3.8,6.7,2.2,Iris-virginica
7.7,2.6,6.9,2.3,Iris-virginica
6.0,2.2,5.0,1.5,Iris-virginica
6.9,3.2,5.7,2.3,Iris-virginica
5.6,2.8,4.9,2.0,Iris-virginica
7.7,2.8,6.7,2.0,Iris-virginica
6.3,2.7,4.9,1.8,Iris-virginica
6.7,3.3,5.7,2.1,Iris-virginica
7.2,3.2,6.0,1.8,Iris-virginica
6.2,2.8,4.8,1.8,Iris-virginica
6.1,3.0,4.9,1.8,Iris-virginica
6.4,2.8,5.6,2.1,Iris-virginica
7.2,3.0,5.8,1.6,Iris-virginica
7.4,2.8,6.1,1.9,Iris-virginica
7.9,3.8,6.4,2.0,Iris-virginica
6.4,2.8,5.6,2.2,Iris-virginica
6.3,2.8,5.1,1.5,Iris-virginica
6.1,2.6,5.6,1.4,Iris-virginica
7.7,3.0,6.1,2.3,Iris-virginica
6.3,3.4,5.6,2.4,Iris-virginica
6.4,3.1,5.5,1.8,Iris-virginica
6.0,3.0,4.8,1.8,Iris-virginica
6.9,3.1,5.4,2.1,Iris-virginica
6.7,3.1,5.6,2.4,Iris-virginica
6.9,3.1,5.1,2.3,Iris-virginica
5.8,2.7,5.1,1.9,Iris-virginica
6.8,3.2,5.9,2.3,Iris-virginica

6.7,3.3,5.7,2.5,Iris-virginica
6.7,3.0,5.2,2.3,Iris-virginica
6.3,2.5,5.0,1.9,Iris-virginica
6.5,3.0,5.2,2.0,Iris-virginica
6.2,3.4,5.4,2.3,Iris-virginica
5.9,3.0,5.1,1.8,Iris-virginica