**Implementasi TinyML untuk Klasifikasi Bunga Iris pada Mikrokontroler ESP32 Menggunakan TensorFlow Lite**

**Mohammad Haryo Hammam Q**

**233140700111108s**

**I. RINGKASAN EKSEKUTIF**

Eksperimen ini mengeksplorasi kemampuan ESP32 sebagai host model pembelajaran mesin ringan (TinyML) untuk mengklasifikasikan jenis bunga Iris berdasarkan fitur morfologis. Model telah dilatih terlebih dahulu, dikonversi ke format TensorFlow Lite, dan kemudian dijalankan pada perangkat ESP32 tanpa bantuan koneksi eksternal maupun sistem operasi. Simulasi dilakukan di platform Wokwi yang memperlihatkan bahwa ESP32 mampu menjalankan prediksi dengan latensi yang rendah dan hasil yang akurat.

**II. LATAR BELAKANG**

Dalam dunia Internet of Things (IoT), permintaan terhadap pengolahan data secara langsung di perangkat (edge computing) semakin meningkat. TinyML menjawab tantangan ini dengan memungkinkan model AI yang ringan dapat berjalan langsung pada mikrokontroler seperti ESP32. Hal ini mengurangi ketergantungan pada server cloud, meningkatkan kecepatan respons, serta menghemat bandwidth dan energi.

Proyek ini menggunakan model klasifikasi dataset *Iris*, salah satu dataset paling ikonik dalam machine learning. Dengan hanya menggunakan empat input numerik — panjang dan lebar sepal serta petal — model dapat membedakan tiga jenis bunga: *Setosa*, *Versicolor*, dan *Virginica*.

**III. TUJUAN**

* Membangun sistem inferensi lokal menggunakan ESP32 dan TensorFlow Lite.
* Memprediksi jenis bunga Iris secara real-time dari empat nilai input.
* Mengukur performa model pada perangkat mikrokontroler (latensi, akurasi).
* Menganalisis potensi TinyML untuk sistem AI berbasis perangkat edge.

**IV. PERANGKAT DAN PERANGKAT LUNAK**

**Perangkat Keras**

* ESP32 Devkit V4 (disimulasikan di Wokwi)

**Perangkat Lunak**

* TensorFlow Lite for Microcontrollers (TFLM)
* Eloquent TinyML Library
* Visual Studio Code + PlatformIO Extension
* Wokwi IoT Simulator
* Dataset Iris (.tflite format)

**V. ARSITEKTUR SISTEM**

**Struktur Input dan Output:**

| **Fitur Masukan** | **Tipe** |
| --- | --- |
| Panjang Sepal | Float |
| Lebar Sepal | Float |
| Panjang Petal | Float |
| Lebar Petal | Float |

Model menghasilkan prediksi kelas dengan probabilitas Softmax sebagai output akhir (kelas 0, 1, atau 2).

**VI. PROSEDUR IMPLEMENTASI**

**1. Pembuatan Proyek**

* Buat proyek baru di PlatformIO pada Visual Studio Code.
* Tambahkan dependensi: eloquent\_tinyml, tflm\_esp32, dan file model .h.

**2. Setup Diagram Wokwi**

Tambahkan ESP32 dan koneksi TX/RX ke serial monitor:

{

"version": 1,

"author": "user",

"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", "", [] ]

],

"dependencies": {}

}

**3. Implementasi Kode Inferensi**

Program utama melakukan prediksi terhadap tiga sampel data dan menampilkan hasil klasifikasi melalui serial monitor.

**Cuplikan Kode:**

#include <Arduino.h>

#include <iris\_model.h>

#include <tflm\_esp32.h>

#include <eloquent\_tinyml.h>

#define ARENA\_SIZE 2000

Eloquent::TF::Sequential<TF\_NUM\_OPS, ARENA\_SIZE> tf;

void setup() {

Serial.begin(115200);

delay(3000);

tf.setNumInputs(4);

tf.setNumOutputs(3);

tf.resolver.AddFullyConnected();

tf.resolver.AddSoftmax();

tf.begin(irisModel);

}

void loop() {

float input1[4] = {5.1, 3.5, 1.4, 0.2};

float input2[4] = {6.0, 2.2, 4.0, 1.0};

float input3[4] = {6.5, 3.0, 5.5, 1.8};

tf.predict(input1);

Serial.print("Hasil Prediksi 1: "); Serial.println(tf.classification);

tf.predict(input2);

Serial.print("Hasil Prediksi 2: "); Serial.println(tf.classification);

tf.predict(input3);

Serial.print("Hasil Prediksi 3: "); Serial.println(tf.classification);

Serial.print("Waktu prediksi: ");

Serial.print(tf.benchmark.microseconds()); Serial.println(" µs");

delay(2000);

}

**VII. HASIL EKSPERIMEN**

**Pengamatan**

| **Sampel** | **Label Seharusnya** | **Hasil Prediksi** | **Waktu (µs)** |
| --- | --- | --- | --- |
| 1 | Setosa (0) | 0 | ~1086 |
| 2 | Versicolor (1) | 1 | ~1086 |
| 3 | Virginica (2) | 2 | ~1086 |

**Analisis**

* Waktu inferensi rata-rata < 1.1 ms, menunjukkan efisiensi komputasi tinggi.
* Model berhasil mengklasifikasikan semua sampel dengan akurasi 100% dalam percobaan ini.
* Tidak ditemukan kesalahan sistem, meski ada peringatan minor dari library.

**VIII. PENUTUP**

**Kesimpulan**

Eksperimen membuktikan bahwa ESP32 mampu menjalankan model klasifikasi sederhana berbasis TensorFlow Lite dengan efisiensi tinggi tanpa koneksi internet. Pendekatan TinyML ini membuka peluang besar untuk penerapan AI pada perangkat edge dengan sumber daya terbatas.