**LAPORAN PRAKTIKUM**

**IOT MINGGU KE-7**

**CLASSIFICATION USING THE TENSORFLOW LITE MODEL ON ESP 32 MICROCONTROLLER**

****

**Dosen Pengampu :**

**Ir. Subairi, ST., MT., IPM**

**Disusun Oleh:**

**Muhammad Alif Aris**

**(233140707111077)**

**PROGRAM STUDI TEKNOLOGI INFORMASI**

**FAKULTAS VOKASI**

**UNIVERSITAS BRAWIJAYA**

**2025**

**Abstrak**

Aplikasi kecerdasan buatan pada mikrokontroler seperti ESP32 telah dimungkinkan berkat peningkatan kemampuan pembelajaran mesin pada perangkat keras yang lebih kecil. Eksperimen ini memeriksa kinerja ESP32 saat menggunakan model TensorFlow Lite untuk klasifikasi data real-time. Selain itu, model pembelajaran mesin yang dioptimalkan diunggah ke ESP32 untuk melakukan inferensi dengan sumber daya terbatas. TensorFlow Lite Micro juga digunakan. Dataset sederhana untuk klasifikasi objek digunakan untuk menguji sistem. Hasil menunjukkan bahwa ESP32 memiliki kemampuan untuk memproses data secara akurat dan menghemat waktu pemrosesan. Selain itu, sistem ini memiliki kemampuan untuk menampilkan hasil inferensi pada Serial Monitor dan mengirimkannya ke platform cloud untuk dianalisis. Eksperimen ini menunjukkan bahwa ESP32 dapat digunakan dalam aplikasi Internet of Things (IoT) berbasis AI, yang menawarkan solusi otomatisasi dan pemantauan lingkungan yang hemat daya.

*Keywords - TensorFlow Lite, ESP32, Klasifikasi Data, Internet of Things, AI Hemat Energi, Pembelajaran Mesin, Inferensi Real-Time.*

**BAB I**

**PENDAHULUAN**

**1.1 Latar Belakang**

Kemajuan dalam pembelajaran mesin dan teknologi mikrokontroler telah membuka peluang baru untuk aplikasi kecerdasan buatan yang dapat dijalankan di perangkat keras kecil seperti ESP32. TensorFlow Lite memungkinkan pembuatan model pembelajaran mesin yang sederhana dan efektif yang dapat dijalankan di perangkat dengan sumber daya terbatas. Dalam eksperimen ini, ESP32 digunakan sebagai platform untuk mengimplementasikan model klasifikasi berbasis TensorFlow Lite. TensorFlow Lite bersama ESP32 memberikan solusi hemat energi untuk aplikasi seperti pengawasan lingkungan, pemantauan kesehatan, dan otomatisasi rumah pintar yang dapat memproses data sensor secara real-time untuk mengklasifikasikan objek atau mendeteksi pola tertentu. Metode ini tidak hanya meningkatkan efisiensi sistem, tetapi juga memungkinkan pengembangan aplikasi Internet of Things yang lebih canggih dan berbasis AI.

**1.2 Tujuan Eksperimen**

1. Untuk pengolahan data lokal, gunakan mikrokontroler ESP32 untuk menerapkan model klasifikasi berbasis TensorFlow Lite.
2. Mengevaluasi kinerja ESP32 saat melakukan inferensi model pembelajaran mesin dengan sumber daya dan daya perangkat yang terbatas.
3. Mengembangkan sistem klasifikasi yang hemat energi untuk aplikasi Internet of Things (IoT) yang beroperasi secara real-time.

**BAB II**

**METODOLOGI**

**2.1 Alat dan Bahan**

* Mikrokontroler ESP32 (Virtual)
* Breadboard (Virtual)
* Kabel Jumper (Virtual)
* Komputer/Laptop dengan akses ke Platform Wokwi
* Dataset atau file model TensorFlow Lite
* Software Arduino IDE atau PlatformIO
* Library TensorFlow Lite untuk ESP32
* Koneksi Internet

**2.3 Implementasi Sistem**

1. Hubungkan ESP32 pada Wokwi dan atur pin input/output untuk model TensorFlow Lite.
2. Unggah model TensorFlow Lite (.tflite) ke ESP32 melalui Arduino IDE.
3. Hubungkan input sensor untuk klasifikasi data sesuai dengan format model.
4. Jalankan inferensi model menggunakan TensorFlow Lite Micro pada ESP32.
5. Tampilkan hasil klasifikasi pada Serial Monitor atau perangkat display.

**BAB III**

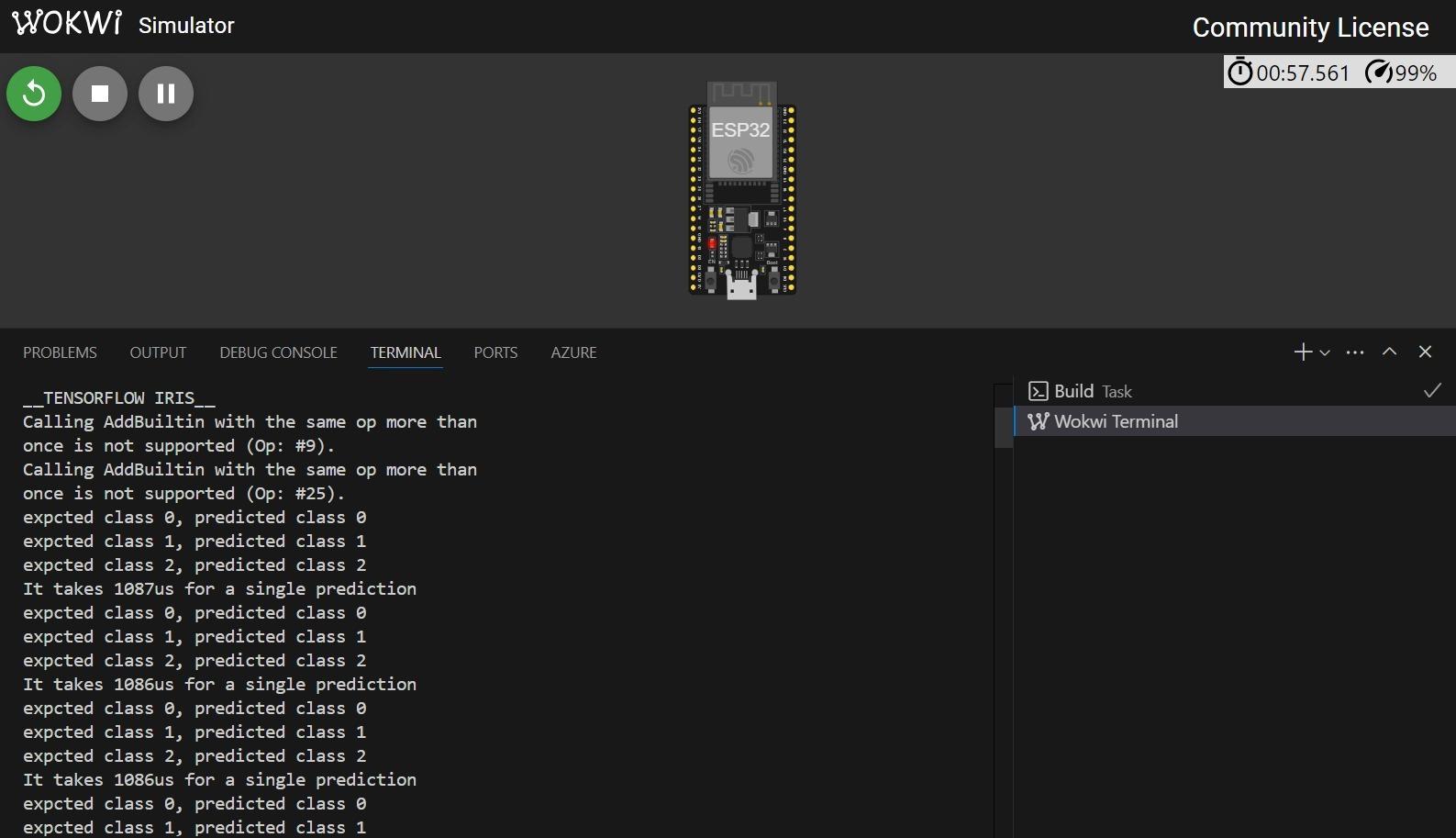
**HASIL DAN PEMBAHASAN**

**3.1 Hasil Eksperimen**

Eksperimen menunjukkan bahwa ESP32 berfungsi dengan baik dengan model TensorFlow Lite untuk klasifikasi data. Ini dioptimalkan untuk perangkat yang memiliki sumber daya terbatas seperti ESP32, yang memungkinkan inferensi dilakukan dengan waktu pemrosesan yang memadai. Data sensor diterima dengan baik. Hasil klasifikasi ditampilkan pada Serial Monitor dan dikirim ke platform cloud untuk analisis.

Meskipun keterbatasan memori ESP32 menjadi masalah, penyesuaian model dan kode memastikan sistem berjalan lancar. Eksperimen ini menunjukkan bahwa ESP32 dapat mendukung model khusus aplikasi IoT berbasis kecerdasan buatan. Dengan integrasi ini, ada kemungkinan pengembangan aplikasi klasifikasi real-time yang hemat energi.

**3.2 Dokumentasi eksperimen meliputi screenshoot simulasi :**

****

**Lampiran**

**Kode Program :**

{

"version": 1,

"author": "subairi",

"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": {}

}

#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);

Serial.println("\_TENSORFLOW IRIS\_");

tf.setNumInputs(4);

tf.setNumOutputs(3);

tf.resolver.AddFullyConnected();

tf.resolver.AddSoftmax();

while (!tf.begin(irisModel).isOk())

Serial.println(tf.exception.toString());

}

void loop() {

if (!tf.predict(x0).isOk()) {

Serial.println(tf.exception.toString());

return;

}

Serial.print("expcted class 0, predicted class ");

Serial.println(tf.classification);

if (!tf.predict(x1).isOk()) {

Serial.println(tf.exception.toString());

return;

}

Serial.print("expcted class 1, predicted class ");

Serial.println(tf.classification);

if (!tf.predict(x2).isOk()) {

Serial.println(tf.exception.toString());

return;

}

Serial.print("expcted class 2, predicted class ");

Serial.println(tf.classification);

Serial.print("It takes ");

Serial.print(tf.benchmark.microseconds());

Serial.println("us for a single prediction");

delay(1000);

}