

LAMPIRAN

Lampiran 1 Surat Kesiediaan Membimbing TA Pembimbing 1

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan di bawah ini :

Nama : Abdul Basit, S.Kom., MT
NIDN : 06.08.12.1991
NIPY : 01.015.198
Jabatan Struktural : LEKTOR
Jabatan Fungsional : KEPALA BAGIAN ADMINISTRASI AKADEMIK

Dengan ini menyatakan bersedia untuk menjadi pembimbing I pada Tugas Akhir mahasiswa berikut :

Nama : Eka Bagas Prasetya
NIM : 23040086
Program Studi : Diploma III Teknik Komputer
Judul TA : KOMPRESI MODEL DEEP NEURAL NETWORK
DENGAN PRUNING DAN QUANTIZATION UNTUK
KLASIFIKASI KUALITAS AIR BERBASIS ESP32

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana semestinya.

Tegal, 20 Februari 2026

Mengetahui

Ka. Prodi DIII Teknik Komputer,

Pembimbing I,


Arif Rakhman, S.E., S.Pd., M.Kom.
NIPY. 05.016.291


Abdul Basit, S.Kom., MT
NIPY. 01.015.198

Lampiran 2 Surat Kesediaan Membimbing TA Pembimbing 2

SURAT KESEDIAN MEMBIMBING TA

Yang bertanda tangan di bawah ini :

Nama : Very Kurnia Bakti, M.Kom
NIDN : 0625118301
NIPY : 09.008.044
Jabatan Struktural : DOSEN TETAP
Jabatan Fungsional : LEKTOR

Dengan ini menyatakan bersedia untuk menjadi pembimbing II pada Tugas Akhir mahasiswa berikut :

Nama : Eka Bagas Prasetya
NIM : 23040086
Program Studi : Diploma III Teknik Komputer
Judul TA : KOMPRESI MODEL DEEP NEURAL NETWORK
DENGAN PRUNING DAN QUANTIZATION UNTUK
KLASIFIKASI KUALITAS AIR BERBASIS ESP32

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana semestinya.

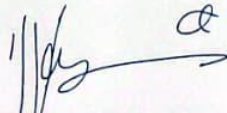
Tegal, 26 Juni 2026

Mengetahui

Ka. Prodi DIII Teknik Komputer,

Pembimbing II,


Arif Rakhman, S.E., S.Pd., M.Kom.
NIPY. 05.016.291


Very Kurnia Bakti, M.Kom
NIPY. 09.008.044

Lampiran 3 Surat Observasi di Perumda Air Minum Tirta Bahari Kota Tegal



Sekolah Vokasi
Program Studi D-3 Teknik Komputer

Nomor : 496.03/KOM-HN/XII/2025
Lampiran : -
Perihal : Permohonan Izin Observasi Tugas Akhir (TA)

Kepada Yth.

Direktur Perumda Air Minum Tirta Bahari Kota Tegal

Di Jl. Hang Tuah No.29, Tegalsari, Kec. Tegal Bar., Kota Tegal, Jawa Tengah 52111

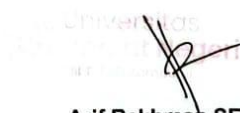
Dengan Hormat,

Sehubungan dengan tugas mata kuliah Tugas Akhir (TA) yang akan diselenggarakan di semester V Program Studi Diploma III Teknik Komputer Universitas Harkat Negeri, Maka dengan ini kami mengajukan izin observasi pengambilan data di Perumda Air Minum Tirta Bahari Kota Tegal yang Bapak / Ibu Pimpin, untuk kepentingan dalam pembuatan produk Tugas Akhir, dengan Mahasiswa sebagai berikut:

No	NIM	NAMA	NO HP
1	23040113	BANI FADHLIH AZKA FEBRIAN	085870582606
2	23040086	EKA BAGAS PRASETYA	083116004590
3	23040078	RIFQI NAUFAL AZIZ	081995347977
4	23040007	FAIZ AHNAF MAULANA	0882008178016

Demikian surat permohonan ini kami sampaikan atas izin dan kerjasamanya kami sampaikan terima kasih.

Tegal, 10 Juni 2026
Ka. Prodi DIII Teknik Komputer
Universitas Harkat Negeri


Arif Rakhman SE, S.Pd. M.Kom
NIPY. 05.016.291

Lampiran 4 Bimbingan Laporan Pembimbing TA

Laporan : Abdul Basit, S.Kom, M.T

No	HARI/ TANGGAL	URAIAN	TANDA TANGAN
1.	26 Mei 2026	Konsul Perubahan judul dan isi	Per
2.	3 Juni 2026	ACC BAB 1-3	
		M. Gus	

PEMBIMBING II: Very Kurnia Bakti, M.Kom BIMBINGAN LAPORAN TA

No	HARI/ TANGGAL	URAIAN	TANDA TANGAN
1.	12 Mei 2026	- Data tidak balance - Akurasi minimal 80 % pada hasil - Menambahkan balancing pada ml	/
2.	19 Mei 2026	- Menambahkan pruning dan kuantisasi untuk optimasi model	
3.	22 Mei 2026	- Menambahkan hasil konfusi matrix pruning & kuantisasi - Mengukur penggunaan energi dan latensi - Mengukur akurasi hasil data ml	/
4.	12 Juni 2026	- Menambahkan metode pengumpulan data (Kaggle)	
5.	15 Juni 2026	- Menampilkan tabel hasil pertandingan model dan, pruning, kuantisasi pd laporan - Menampilkan hasil pengukuran pada alat	/
6.	19 Juni 2026	- Uji coba alat (merupakan hasil akurasi pengukuran sensor berdasarkan data) - Perubahan kata pada judul	
7.	1 April 2026	- Revisi Judul - Model ML ganti mencari akurasi tertinggi	/

No	HARI/ TANGGAL	URAIAN	TANDA TANGAN
8.	26 Juni 2026	- Menambahkan hasil penggunaan memori ESP32 <u>Siap di uji bu</u> Dr, ~~26~~ 29/6-2026	h 

Lampiran 5 Dokumentasi Hasil Observasi



Lampiran 6 Source Code

```
#include <Wire.h>
#include <Adafruit_INA219.h>

#include "tensorflow/lite/micro/micro_interpreter.h"
#include "tensorflow/lite/micro/micro_mutable_op_resolver.h"
#include "tensorflow/lite/schema/schema_generated.h"

#include "model.h" // Membaca model biner INT8 Anda
#include "dataset.h" // Membaca data uji terstruktur secara otomatis
dari tab dataset.h

// Inisialisasi Sensor Daya INA219
Adafruit_INA219 ina219;

// Alokasi memori statis (40 KB) untuk kelancaran komputasi
constexpr int kTensorArenaSize = 40 * 1024;
alignas(16) uint8_t tensor_arena[kTensorArenaSize];

// Variabel Utama TensorFlow Lite Micro
const tflite::Model* tflite_model = nullptr;
tflite::MicroInterpreter* interpreter = nullptr;
TfLiteTensor* input_tensor = nullptr;
TfLiteTensor* output_tensor = nullptr;

// Indeks pengujian batch
int current_index = 0;
bool testing_done = false;

// Variabel untuk menghitung akurasi on-chip
int total_benar = 0;

void setup() {
    Serial.begin(115200);

    // Inisialisasi komunikasi I2C (SDA = Pin 21, SCL = Pin 22)
    Wire.begin(21, 22);
    delay(2000);

    Serial.println("\n=====
==");
    Serial.println("  EVALUASI REKAYASA TINYML - BATCH DATASET
FIXED  ");
```

```

    Serial.println("  SISTEM KLASIFIKASI KUALITAS AIR PERUMDA
TEGAL  ");
    Serial.println("=====
");

    // Cek Koneksi Sensor INA219
    if (!ina219.begin()) {
        Serial.println("[ERROR] Modul INA219 tidak merespon. Periksa pin
kabel!");
        while (1) { delay(100); }
    }
    ina219.setCalibration_32V_2A();
    Serial.println("[OK] Sensor Telemetri INA219 Terdeteksi.");

    // Memuat Model dari file model.h
    tflite_model = tflite::GetModel(model);
    if (tflite_model->version() != TFLITE_SCHEMA_VERSION) {
        Serial.println("[ERROR] Versi schema model .h tidak cocok!");
        while (1) { delay(100); }
    }
    Serial.println("[OK] Model INT8 Berhasil Dimuat.");

    // Mendaftarkan Layer Operator (Resolver)
    static tflite::MicroMutableOpResolver<3> resolver;
    resolver.AddFullyConnected();
    resolver.AddRelu();
    resolver.AddLogistic();

    // Membangun Interpreter
    static tflite::MicroInterpreter static_interpreter(
        tflite_model, resolver, tensor_arena, kTensorArenaSize);
    interpreter = &static_interpreter;

    // Alokasi Tensor Memori
    TfLiteStatus status = interpreter->AllocateTensors();
    if (status != kTfLiteOk) {
        Serial.println("[ERROR] Alokasi RAM Tensor Arena Gagal!");
        while (1) { delay(100); }
    }

    input_tensor = interpreter->input(0);
    output_tensor = interpreter->output(0);

    Serial.print("[OK] Alokasi Tensor Selesai. Total Data Terdeteksi:
");

```

```

Serial.println(DATASET_SIZE);

// MENAMPILKAN MEMORI AWAL SAAT STARTUP (Tambahan untuk Evaluasi)
Serial.print("[INFO] Sisa RAM Awal (Free Heap): ");
Serial.print(ESP.getFreeHeap() / 1024); // Konversi byte ke KB
Serial.println(" KB");
Serial.println("=====
\n");
}

void loop() {
    // Jika seluruh dataset selesai dievaluasi, tampilkan rangkuman
    akhir akurasi
    if (testing_done) {
        Serial.println("\n=====
=====");
        Serial.print(">> [SUKSES] Seluruh ");
        Serial.print(DATASET_SIZE);
        Serial.println(" Dataset Selesai Dievaluasi Sempurna. <<");
        Serial.print("    TOTAL AKURASI ALAT ON-CHIP: ");
        Serial.print(((float)total_benar / DATASET_SIZE) * 100.0f, 2);
        Serial.println("%");

        // MENAMPILKAN SISA RAM AKHIR UNTUK DATA LAPORAN (Tambahan untuk
        Evaluasi)
        Serial.print("    SISA RAM AKHIR (FREE HEAP) : ");
        Serial.print(ESP.getFreeHeap() / 1024);
        Serial.println(" KB");
        Serial.println("=====
=====");

        delay(30000); // Diam selama 30 detik sebelum mengulang log
        akurasi
        return;
    }

    // 1. Mengambil Nilai Sensor dan Kunci Jawaban dari dataset.h
    float pH = test_dataset[current_index][0];
    float TDS = test_dataset[current_index][1];
    float Turbidity = test_dataset[current_index][2];
    int8_t label_asli = ground_truth[current_index];

    // 2. Proses Kuantisasi Input secara Aman ke INT8
    float input_scale = input_tensor->params.scale;
    int32_t input_zero_point = input_tensor->params.zero_point;

```

```

    if (input_scale == 0) input_scale = 1.0f;

    int32_t q_pH      = (int32_t)(pH / input_scale) +
input_zero_point;
    int32_t q_TDS      = (int32_t)(TDS / input_scale) +
input_zero_point;
    int32_t q_Turbidity = (int32_t)(Turbidity / input_scale) +
input_zero_point;

    input_tensor->data.int8[0] = (int8_t)constrain(q_pH, -128, 127);
    input_tensor->data.int8[1] = (int8_t)constrain(q_TDS, -128, 127);
    input_tensor->data.int8[2] = (int8_t)constrain(q_Turbidity, -128,
127);

    // 3. Menghitung Latensi Waktu Inferensi AI
    uint32_t start_time = micros();
    TfLiteStatus invoke_status = interpreter->Invoke();
    uint32_t end_time = micros();

    if (invoke_status != kTfLiteOk) {
        Serial.println("[ERROR] Perintah Invoke AI Gagal!");
        return;
    }
    float latency_us = end_time - start_time;

    // 4. Membaca Konsumsi Daya Riil dari INA219
    float voltage_V = ina219.getBusVoltage_V();
    float current_mA = ina219.getCurrent_mA();
    float power_mW = ina219.getPower_mW();

    // 5. Pembacaan Hasil Berdasarkan Raw INT8 (Mengatasi Mismatch
Model)
    int8_t raw_output = output_tensor->data.int8[0];
    int32_t output_zero_point = output_tensor->params.zero_point;

    // Logika Klasifikasi Berdasarkan Pergeseran Titik Nol Kuantisasi
Model
    String status_air;
    int8_t label_prediksi = 0;

    if (raw_output > output_zero_point) {
        status_air = "LAYAK KONSUMSI";
        label_prediksi = 1;
    } else {
        status_air = "TIDAK LAYAK KONSUMSI";
    }

```

```

    label_prediksi = 0;
}

// Validasi kecocokan on-chip dengan Kunci Jawaban Google Colab
String validasi;
if (label_prediksi == label_asli) {
    validasi = "SESUAI (VALID)";
    total_benar++;
} else {
    validasi = "TIDAK SESUAI (MISMATCH)";
}

// 6. Cetak Log Parameter, Keputusan Langsung, dan Telemetri Daya
Serial.print("[DATA KE-"); Serial.print(current_index + 1);
Serial.print("/"); Serial.print(DATASET_SIZE); Serial.println("]");

Serial.print("    PARAMETER    -> pH: "); Serial.print(pH, 4);
Serial.print(" | TDS: "); Serial.print(TDS, 4);
Serial.print(" | Turbidity: "); Serial.println(Turbidity, 4);

Serial.print("    KEPUTUSAN    -> "); Serial.print(status_air);
Serial.print(" | Validasi Alat vs Colab: ");
Serial.println(validasi);

Serial.print("    TELEMETRI    -> Latensi: ");
Serial.print(latency_us, 0); Serial.print(" us");
Serial.print(" | Tegangan: "); Serial.print(voltage_V);
Serial.print(" V");
Serial.print(" | Arus: "); Serial.print(current_mA);
Serial.print(" mA");
Serial.print(" | Daya: "); Serial.print(power_mW);
Serial.println(" mW");
Serial.println("-----");
Serial.println("-----");

// 7. Logika Delay Konstan (Kecil) dan Jeda Istirahat Setiap 500
Data
delay(5);

// Cek apakah data yang baru diproses adalah kelipatan 500
if ((current_index + 1) % 500 == 0) {
    Serial.println("\n=====");
    Serial.print("[INFO] Berhasil mengevaluasi ");
    Serial.print(current_index + 1);

```

```

Serial.println(" data.");

// MENAMPILKAN SISA RAM SECARA BERKALA (Tambahan untuk Evaluasi
Jeda)
Serial.print("[MEMORI] Sisa RAM saat ini (Free Heap): ");
Serial.print(ESP.getFreeHeap() / 1024);
Serial.println(" KB");

Serial.println("[COOLING DOWNS] Berhenti selama 10 detik untuk
mencegah kenaikan suhu chip (Overheat)...");
Serial.println("=====
=====\\n");

delay(10000); // Istirahat total selama 10 detik
}

// Naikkan indeks untuk data berikutnya
current_index++;

// Cek jika batas akhir total dataset sudah tercapai
if (current_index >= DATASET_SIZE) {
    testing_done = true;
}
}

```