

LAMPIRAN

Lampiran 1 Surat Kesiediaan Pembimbing 1 Tugas Akhir

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan dibawah ini:

Nama	: Very Kurnia Bakti, M.Kom.
NIDN	: 0625118301
NIPY	: 09.008.044
Jabatan Struktural	: Dosen Tetap
Jabatan Fungsional	: Dosen Tetap

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

Nama	: Ahmad Sofwan Alfin
NIM	: 23041034
Program Studi	: D-3 Teknik Komputer
Judul TA	: IMPLEMENTASI <i>TINY MACHINE LEARNING</i> BERBASIS <i>MULTI-LAYER PERCEPTRON</i> PADA ESP32-S3 UNTUK DETEKSI ANOMALI GETARAN KOMPRESOR TRUK BERPENDINGIN

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana mestinya.

Tegal, 20 Februari 2026

Mengetahui

Ka. Prodi DIII Teknik Komputer,

Dosen Pembimbing I,


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


Very Kurnia Bakti, M.Kom.
NIPY. 09.008.044

Lampiran 2 Surat Kesiediaan Pembimbing 2 Tugas Akhir

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan dibawah ini:

Nama	: Abdul Basit, S.Kom., MT.
NIDN	: 06.08.12.1991
NIPY	: 01.015.198
Jabatan Struktural	: Sekretaris Prodi
Jabatan Fungsional	: Lektor

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

Nama	: Ahmad Sofwan Alfin
NIM	: 23041034
Program Studi	: D-3 Teknik Komputer
Judul TA	: IMPLEMENTASI <i>TINY MACHINE LEARNING</i> BERBASIS <i>MULTI-LAYER PERCEPTRON</i> PADA ESP32-S3 UNTUK DETEKSI ANOMALI GETARAN KOMPRESOR TRUK BERPENDINGIN

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana mestinya.

Tegal, 20 Februari 2026

Mengetahui

Ka. Prodi DIII Teknik Komputer,

Dosen Pembimbing II,


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


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

Lampiran 3 Surat Balasan Observasi



CV. MUTIARA SAMUDERA INDONESIA (MSI)

Jl. Bali No.35, Mintaragen, Kec. Tegal Tim., Kota Tegal, Jawa Tengah 52121

Nomor : 001/MSI/IV/2026

Tegal, 08 April 2026

Lampiran : -

Perihal : Persetujuan Izin Observasi Tugas Akhir

Kepada Yth.

Ka. Prodi D-III Teknik Komputer

Universitas Harkat Negeri

di Tempat

Dengan hormat,

Menindaklanjuti surat Nomor **579.03/KOM-HN/IV/2026** tanggal **07 April 2026** perihal **Permohonan Izin Observasi Tugas Akhir (TA)**, kami menyampaikan bahwa **CV Mutiara Samudera Indonesia** memberikan izin kepada mahasiswa Program Studi D-III Teknik Komputer Universitas Harkat Negeri yang namanya tercantum di bawah ini untuk melaksanakan kegiatan observasi dan pengambilan data di perusahaan kami:

No	Nama	Nim
1	Luckman Hasanudin	23040029
2	Citra Sholila	23040041
3	Nur Sofiatun Nisa	23041030
4	Ahmad Sofwan Alfin	23041034

Kegiatan observasi tersebut dapat dilaksanakan sesuai dengan ketentuan dan peraturan yang berlaku di lingkungan perusahaan. Data yang diperoleh diharapkan digunakan hanya untuk kepentingan akademik dalam penyusunan Tugas Akhir. Demikian surat persetujuan ini kami sampaikan. Atas perhatian dan kerja sama yang baik, kami ucapkan terima kasih.

Tegal, 08 April 2026

Hormat kami,

CV Mutiara Samud



Lampiran 4 Potongan Kode *Python* Pelatihan Model

```
1  """
2  2_train.py
3  =====
4  9 fitur -> MLP Dense(32,16) -> TFLite Float32
5  """
6
7  import NumPy as np
8  import pandas as pd
9  import TensorFlow as tf
10 import joblib
11 from sklearn.model_selection import train_test_split
12 from sklearn.preprocessing import StandardScaler, LabelEncoder
13 from sklearn.metrics import classification_report, confusion_matrix
14
15 FEATURES_FILE = "features.csv"
16 EPOCHS        = 150
17 BATCH_SIZE    = 64
18 TEST_SIZE     = 0.2
19 RANDOM_SEED   = 42
20 LABELS        = ["Bearing", "Imbalance", "Normal"]
21
22 FEATURE_COLS = [
23     "RMS_X", "RMS_Y", "RMS_Z",
24     "Peak_X", "Peak_Y", "Peak_Z",
25     "Energy_X", "Energy_Y", "Energy_Z",
26 ]
27
28 # --- 1. LOAD DATA -----
29 print("=" * 50)
30 print("1. Load data...")
31 df = pd.read_csv(FEATURES_FILE)
32 X = df[FEATURE_COLS].values.astype(np.Float32)
33 y = df["Label"].values
34 print(f" X shape : {X.shape}")
35 for name in LABELS:
36     print(f" {name:10s} : {np.sum(y==name):>10,} sampel")
37
38 df[["Label"] + FEATURE_COLS].to_csv("dataset_fitur_9.csv", index=False)
39 print(" Tersimpan: dataset_fitur_9.csv")
40
41 # --- 2. LABEL ENCODER -----
42 print("\n2. Label Encoder...")
43 le = LabelEncoder()
44 y_encoded = le.fit_transform(y)
45 joblib.dump(le, "label_encoder_v4.save")
46 print(f" Kelas: {list(le.classes_)}")
47
48 # --- 3. NORMALISASI -----
49 print("\n3. Normalisasi...")
50 scaler = StandardScaler()
51 X_scaled = scaler.fit_transform(X).astype(np.Float32)
52 joblib.dump(scaler, "scaler_v4.save")
53 mean_val = scaler.mean_.astype(np.Float32)
54 scale_val = scaler.scale_.astype(np.Float32)
55
56 # --- 4. SPLIT TRAIN/TEST -----
57 print("\n4. Split train/test...")
58 X_train, X_test, y_train, y_test = train_test_split(
59     X_scaled, y_encoded, test_size=TEST_SIZE
```

```

60     random_state=RANDOM_SEED, stratify=y_encoded)
61 print(f"    Train : {X_train.shape[0]:,}")
62 print(f"    Test  : {X_test.shape[0]:,}")
63
64 # --- 5. BUILD MODEL -- Dense(32,16) -----
65 print("\n5. Build model MLP Dense(32,16)...")
66 model = tf.keras.Sequential([
67     tf.keras.layers.Input(shape=(9,)),
68     tf.keras.layers.Dense(32, activation="relu"),
69     tf.keras.layers.Dense(16, activation="relu"),
70     tf.keras.layers.Dense(3, activation="Softmax"),
71 ], name="v1_base_f32")
72 model.compile(
73     optimizer="adam",
74     loss="sparse_categorical_crossentropy",
75     metrics=["accuracy"],
76 )
77 model.summary()
78
79 # --- 6. TRAINING -----
80 print("\n6. Training...")
81 callbacks = [
82     tf.keras.callbacks.EarlyStopping(patience=15, restore_best_weights=True),
83     tf.keras.callbacks.ReduceLROnPlateau(patience=7, factor=0.5, verbose=1),
84 ]
85 model.fit(
86     X_train, y_train,
87     validation_data=(X_test, y_test),
88     epochs=EPOCHS,
89     batch_size=BATCH_SIZE,
90     callbacks=callbacks,
91     verbose=1,
92 )
93
94 # --- 7. EVALUASI -----
95 print("\n7. Evaluasi...")
96 loss, acc = model.evaluate(X_test, y_test, verbose=0)
97 print(f"    Test accuracy : {acc:.4f}")
98 y_pred = np.argmax(model.predict(X_test, verbose=0), axis=1)
99 print(classification_report(y_test, y_pred, target_names=LABELS))
100 print(confusion_matrix(y_test, y_pred))
101
102 # --- 7b. SIMPAN MODEL KERAS (dipakai V2 untuk fine-tune pruning) -----
103 model.save("v1_base_f32.keras")
104 print("    Tersimpan: v1_base_f32.keras")
105
106 # --- 8. EXPORT TFLITE FLOAT32 -----
107 print("\n8. Export TFLite Float32...")
108 converter = tf.lite.TFLiteConverter.from_keras_model(model)
109 TFLite_model = converter.convert()
110 with open("v1_base_f32.TFLite", "wb") as f:
111     f.write(TFLite_model)
112 print(f"    Tersimpan: v1_base_f32.TFLite ({len(TFLite_model):,} bytes)")
113
114 # --- 9. CONVERT KE basef32.h -----
115 print("\n9. Convert ke basef32.h...")
116 hex_array = ", ".join(f"0x{b:02x}" for b in TFLite_model)
117 header = (

```

```

118     "#ifndef BASEF32_H\n#define BASEF32_H\n\n"
119     f"const unsigned char basef32_data[] = {{\n  {hex_array}\n}};\n"
120     f"const unsigned int basef32_data_len = {len(TFLite_model)};\n\n"
121     "#endif // BASEF32_H\n"
122 )
123 with open("basef32.h", "w") as f:
124     f.write(header)
125 print("    Tersimpan: basef32.h")

```

Lampiran 5 Potongan Kode Implementasi *TinyML* Pada ESP32S3

```

1 // =====
2 // TAHAP 1: INISIALISASI SISTEM (Konfigurasi Sensor MPU6050)
3 // =====
4 if (mpu.begin(0x69)) {
5     mpuOK = true;
6     mpu.setAccelerometerRange(MPU6050_RANGE_8_G);
7     mpu.setFilterBandwidth(MPU6050_BAND_94_HZ);
8
9     // =====
10    // TAHAP 2: KALIBRASI OFFSET (200 sampel kondisi diam)
11    // =====
12    float sX = 0, sY = 0, sZ = 0;
13    for (int i = 0; i < 200; i++) {
14        sensors_event_t a, g, t;
15        mpu.getEvent(&a, &g, &t);
16        sX += a.acceleration.x;
17        sY += a.acceleration.y;
18        sZ += a.acceleration.z;
19        vTaskDelay(5 / portTICK_PERIOD_MS);
20    }
21    offX = sX / 200.0f;
22    offY = sY / 200.0f;
23    offZ = sZ / 200.0f;
24
25    // Koreksi komponen gravitasi pada sumbu dominan
26    float absX = fabs(offX), absY = fabs(offY), absZ = fabs(offZ);
27    if (absY >= absX && absY >= absZ)
28        offY -= (offY > 0 ? 9.81f : -9.81f);
29    else if (absZ >= absX && absZ >= absY)
30        offZ -= (offZ > 0 ? 9.81f : -9.81f);
31    else
32        offX -= (offX > 0 ? 9.81f : -9.81f);
33 }
34
35 // =====
36 // TAHAP 3: AKUISISI DATA (128 sampel @ 200 Hz) + NORMALISASI
37 // =====
38 for (int i = 0; i < WINDOW_SIZE; i++) {
39     unsigned long ts_us = micros();
40     sensors_event_t a, g, t;
41     if (mpuOK && mpu.getEvent(&a, &g, &t)) {
42         ax_buf[i] = (a.acceleration.x - offX) / 9.81f; // normalisasi ke
satuan-g
43         ay_buf[i] = (a.acceleration.y - offY) / 9.81f;
44         az_buf[i] = (a.acceleration.z - offZ) / 9.81f;
45     } else {
46         ax_buf[i] = ay_buf[i] = az_buf[i] = 0.0f;
47     }
48     while (micros() - ts_us < SAMPLING_PERIOD_US)
49         yield();
50 }
51
52 // =====
53 // TAHAP 4: BANDPASS FILTER (Butterworth orde 4, 5-90 Hz)
54 // =====
55 resetFilterState();
56 applyBandpassFilter(ax_buf, 0);
57 applyBandpassFilter(ay_buf, 1);
58 applyBandpassFilter(az_buf, 2);

```

```

59
60 // =====
61 // TAHAP 5: WINDOWING + FFT + EKSTRAKSI FITUR SPEKTRAL
62 // =====
63 void computeFFTFeatures(float *input, float &peak, float &energy) {
64     float avg = 0.0f;
65     for (int i = 0; i < WINDOW_SIZE; i++) avg += input[i];
66     avg /= WINDOW_SIZE;
67
68     for (int i = 0; i < WINDOW_SIZE; i++) {
69         vReal[i] = input[i] - avg; // hilangkan komponen DC
70         vImag[i] = 0.0f;
71     }
72
73     FFT.windowing(vReal, WINDOW_SIZE, FFT_WIN_TYP_HANN, FFT_FORWARD);
74     FFT.compute(vReal, vImag, WINDOW_SIZE, FFT_FORWARD);
75     FFT.complexToMagnitude(vReal, vImag, WINDOW_SIZE);
76
77     peak = 0.0f; energy = 0.0f;
78     for (int i = 1; i < WINDOW_SIZE / 2; i++) {
79         float val = vReal[i] / WINDOW_SIZE; // normalisasi terhadap N
80         if (val > peak) peak = val;
81         energy += val * val;
82     }
83 }
84
85 void extractFeatures(float feat[9]) {
86     float px, py, pz, ex, ey, ez;
87     computeFFTFeatures(ax_buf, px, ex);
88     computeFFTFeatures(ay_buf, py, ey);
89     computeFFTFeatures(az_buf, pz, ez);
90     feat[0] = computeRMS(ax_buf);
91     feat[1] = computeRMS(ay_buf);
92     feat[2] = computeRMS(az_buf);
93     feat[3] = px; feat[4] = py; feat[5] = pz; // Peak Amplitude
94     feat[6] = ex; feat[7] = ey; feat[8] = ez; // Spectral Energy
95 }
96
97 // =====
98 // TAHAP 6: INFERENSI MLP (TFLite Micro)
99 // =====
100 for (int i = 0; i < 9; i++) {
101     float norm = (features[i] - mean_val[i]) / scale_val[i];
102     input_tensor->data.f[i] = constrain(norm, -5.0f, 5.0f);
103 }
104 interpreter->Invoke();
105
106 int raw_pred = 0;
107 float max_prob = -1.0f;
108 for (int i = 0; i < 3; i++) {
109     float p = output_tensor->data.f[i];
110     if (p > max_prob) { max_prob = p; raw_pred = i; }
111 }
112
113 // =====
114 // TAHAP 7: HASIL KLASIFIKASI KONDISI MESIN
115 // =====
116 int current_label = (max_prob < CONFIDENCE_THRESHOLD) ? 3 : raw_pred;
117 // current_label -> LABELS[]: "Bearing", "Imbalance", "Normal", "Unknown"

```

Lampiran 6 Foto Dokumentasi Observasi dan Wawancara

