

LAMPIRAN

Lampiran 1. Surat Kesiediaan Dosen Pembimbing I

SURAT KESEDIAN 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 : Bani Fadhlih Azka Febrian
NIM : 23040113
Program Studi : Diploma III Teknik Komputer
Judul TA : IMPLEMENTASI *DEEP NEURAL NETWORK* (DNN)
DENGAN DATA *FUSION* UNTUK KLASIFIKASI
KUALITAS AIR

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 Dosen Pembimbing II

SURAT KESEDIAAN 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 : Bani Fadhlih Azka Febrian
NIM : 23040113
Program Studi : Diploma III Teknik Komputer
Judul TA : IMPLEMENTASI *DEEP NEURAL NETWORK* (DNN)
DENGAN DATA *FUSION* UNTUK KLASIFIKASI
KUALITAS AIR

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana semestinya.

Tegal, 20 Februari 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

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

Program Studi D-3 Teknik Komputer Sekolah Vokasi
Gedung Utama Ruang B.09 Kampus Mataram - Jalan Mataram No.9 Kota Tegal
WhatsApp: +62-85291833673 | Surel: komputer@harkatnegeri.ac.id

Dipindai dengan CamScanner

Lampiran 4. Bimbingan Laporan Pembimbing I

Laporan : Abdul Basit, S.Kom. AMT

No	HARI/ TANGGAL	URAIAN	TANDA TANGAN
1.	3 Juni 2026	BAB I Latar belakang ada satu kata yang ambigu	ay
2.	26 Mei 2026	-BAB III Prosedur Penelitian masih salah harus sesuai dengan alur training ML, -BAB IV dan V membahas tentang proses selama training ataupun testing dari ML (saran)	CU

Basit 1.2.5

Lampiran 5. Bimbingan Laporan Pembimbing II

Lampiran 23
Bimbingan Laporan Pembimbing I TA

IK P2M PHB d.5.1.e.1

PEMBIMBING II: Verry Kurnia, M.Kom BIMBINGAN LAPORAN TA

No	HARI/ TANGGAL	URAIAN	TANDA TANGAN
1.	1 April 2026	- Membuat sequence diagram - Revisi Judul - Mencoba training menggunakan regresi linier KNN dan smother - Melakukan pre-processing	h
2.	20 April 2026	- Fokus ML di Perangkat IoT - Model ML diubah agar akurasi bisa mencapai 80% - 90% - Kodungan ML masih perlu perbaikan, split data dilakukan sebelum augmentasi.	h
3.	12 Juni 2026	- Kalimat awal tidak boleh asing - Teori terkait - tambahkan library yang digunakan pada landasan teori	h

No	HARI/ TANGGAL	URAIAN	TANDA TANGAN
	12 Juni 2026	- Pruning dan quantization tidak perlu ditambahkan - Optimasi model dihapus - Epoch diganti menjadi 100 yang sebelumnya 50 - Menambahkan rumus pada preprocessing - Preprocessing terlebih dahulu sebelum yang lainnya.	h
4.	19 Juni 2026	- Hasil - Confusion Matrix gambar lebih diperjelas - Hasil konversi menampilkan ukuran.	h
5.	26 Juni 2026		

No	HARI/ TANGGAL	URAIAN	TANDA TANGAN
6.	12 Mei 2026	- Cleaning data anomali dan SMOTE - Menambahkan dataset	h
7.	19 Mei 2026	- Menambahkan optimasi model	
8.	22 Mei 2026	- Menampilkan hasil tambahan dataset	
Grup Uti			26/6/2026

Lampiran 6. Dokumentasi



Lampiran 7. *Source Code* Program

```
# TRAININGG

import os
import random
import joblib
import numpy as np
import pandas as pd
import tensorflow as tf
import matplotlib.pyplot as plt
import seaborn as sns

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import (
    accuracy_score,
    classification_report,
    confusion_matrix
)

from imblearn.over_sampling import SMOTE

#
=====
# RANDOM SEED
#
=====

np.random.seed(42)
tf.random.set_seed(42)
random.seed(42)

#
=====
# LOAD DATASET
#
=====
```

```

print("=====")
print("LOAD DATASET")
print("=====")

df_19k = pd.read_csv(
    'waterquality_labeled.csv'
)

df_3k = pd.read_csv(
    'water_quality.csv'
)

df_macao = pd.read_csv(
    'Macao_Water_Quality_Dataset_Original_dataset.csv'
)

print(
    f"19K : {df_19k.shape}"
)

print(
    f"3K : {df_3k.shape}"
)

print(
    f"Macao : {df_macao.shape}"
)

#
=====
# RENAME
#
=====

df_19k=df_19k.rename(
    columns={
        'Potability':'potability'
    }
)

df_macao=df_macao.rename(
    columns={

```

```

        'pH':'ph',
        'TDS':'tds',
        'Turbidity':'turbidity'
    }
)

#
=====
#  AMBIL KOLOM
#
=====

df_19k=df_19k[
[
'ph',
'tds',
'turbidity',
'potability'
]
]

df_3k=df_3k[
[
'ph',
'tds',
'turbidity',
'potability'
]
]

#
=====
#  AUTO LABEL MACAO
#
=====

df_macao=df_macao[
[
'ph',
'tds',
'turbidity'
]
]

```

```

]

def create_label(row):

    if (

        6.5<=row['ph']<=8.5
        and
        row['tds']<=500
        and
        row['turbidity']<=5

    ):

        return 1

    return 0

df_macao['potability']=df_macao.apply(

    create_label,
    axis=1
)

#
=====
# HAPUS NULL
#
=====

df_19k=df_19k.dropna()
df_3k=df_3k.dropna()
df_macao=df_macao.dropna()

print("\nClean selesai")

#
=====
# GABUNGAN DATASET DENGAN BATAS MACAO
#
=====

```

```

print("\n=====")
print(" GABUNGAN DATASET ")
print("=====")

#
=====
# HITUNG TOTAL DATA ASLI
#
=====

total_real = (

    len(df_19k)
    +
    len(df_3k)

)

print(
f"\nTotal data asli: {total_real}"
)

#
=====
# MACAO HANYA 30%
#
=====

macao_size = int(
    total_real*0.3
)

print(
f"Macao digunakan: {macao_size}"
)

if len(df_macao)>macao_size:

    df_macao=df_macao.sample(

        n=macao_size,

```

```

        random_state=42
    )

#
=====
#  GABUNG
#
=====

df=pd.concat([

df_19k[
[
'ph',
'tds',
'turbidity',
'potability'
]
],

df_3k[
[
'ph',
'tds',
'turbidity',
'potability'
]
],

df_macao[
[
'ph',
'tds',
'turbidity',
'potability'
]
]

])

```

```

#
=====
# SHUFFLE
#
=====

df=df.sample(

frac=1,

random_state=42

).reset_index(
drop=True
)

print("\nDataset final:")
print(df.shape)

print("\nDistribusi label:")

print(
df['potability'].value_counts()
)

#
=====
# FITUR DAN LABEL
#
=====

X=df[
[
'ph',
'tds',
'turbidity'
]
]

y=df[
'potability'
]

```

```

#
=====
# TRAIN TEST SPLIT
#
=====

X_train,X_test,y_train,y_test=train_test_split(

X,
Y,

test_size=0.2,

random_state=42,

stratify=y
)

print(
f"\nTraining:{len(X_train)}"
)

print(
f"Testing:{len(X_test)}"
)

#
=====
# STANDARD SCALER
#
=====

scaler=StandardScaler()

X_train_scaled=scaler.fit_transform(
X_train
)

X_test_scaled=scaler.transform(
X_test
)

```

```

#
=====
# SMOTE
#
=====

smote=SMOTE(
random_state=42
)

X_train_smote,y_train_smote=smote.fit_resample(

X_train_scaled,
y_train
)

print("\nSetelah SMOTE:")

print(
np.bincount(
y_train_smote
)
)

#
=====
# BUILD DNN
#
=====

model=Sequential([

Dense(
16,
activation='relu',
input_shape=(3,)
),

Dense(
8,
activation='relu'
),

```

```

Dense(
1,
activation='sigmoid'
)

])

model.compile(

optimizer=Adam(
learning_rate=0.001
),

loss='binary_crossentropy',

metrics=['accuracy']
)

early_stop=EarlyStopping(
monitor='val_loss',

patience=5,

restore_best_weights=True
)

#
=====
# TRAINING
#
=====

history=model.fit(

X_train_smote,

y_train_smote,

validation_split=0.2,

epochs=100,

```

```

batch_size=128,

callbacks=[early_stop],

verbose=1
)

#
=====
# TESTING
#
=====

y_pred_prob=model.predict(
X_test_scaled
)

y_pred=(

y_pred_prob>0.5

).astype(int)

accuracy=accuracy_score(

y_test,
y_pred
)

print(
f"\nAccuracy:{accuracy*100:.2f}%"
)

print(

classification_report(

y_test,
y_pred,

target_names=[

```

```

'Tidak Layak',

'Layak'

]

))

cm=confusion_matrix(
y_test,
y_pred
)

plt.figure(
figsize=(8,6)
)

sns.heatmap(

cm,

annot=True,

fmt='d',

cmap='Blues'
)

plt.title(

f'DNN ({accuracy*100:.2f}%)'
)
plt.savefig(
    'confusion_matrix_training.png',
    dpi=300,
    bbox_inches='tight'
)
plt.show()

#
=====
# SAVE DATASET FINAL

```

```

#
=====

print("\n=====")
print(" SAVE DATASET ")
print("=====")

os.makedirs(
    'ml/dataset',
    exist_ok=True
)

df.to_csv(
    'ml/dataset/dataset_final_realistic.csv',
    index=False
)

print(
    "\nDataset final tersimpan"
)

print(
    f"Total data: {len(df)}"
)

#
=====
# SAVE MODEL DNN
#
=====

print("\n=====")
print(" SAVE MODEL ")
print("=====")

os.makedirs(
    'ml/model',
    exist_ok=True
)

# format keras (disarankan)
model.save(

```

```

        'ml/model/dnn_final_realistic.keras'
    )

    print(
        "\nModel DNN tersimpan"
    )

    #
    =====
    # CONVERT TFLITE
    #
    =====

    converter = tf.lite.TFLiteConverter.from_keras_model(
        model
    )

    tflite_model = converter.convert()

    with open(
        'ml/model/dnn_final_realistic.tflite',
        'wb'
    ) as f:

        f.write(
            tflite_model
        )

    print(
        "\nModel TFLite tersimpan"
    )

    #
    =====
    # SAVE SCALER
    #
    =====

    joblib.dump(
        scaler,

```

```

        'ml/model/scaler_realistic.save'
    )

    print(
        "\nScaler tersimpan"
    )

    #
    =====
    # SAVE TEST RESULT
    #
    =====

    hasil = X_test.copy()

    hasil['actual'] = y_test.values

    hasil['prediction'] = y_pred.flatten()

    hasil['confidence'] = (
        y_pred_prob.flatten()*100
    )

    os.makedirs(
        'ml/testing',
        exist_ok=True
    )

    hasil.to_csv(

        'ml/testing/hasil_test_realistic.csv',

        index=False
    )

    print(
        "\nHasil testing tersimpan"
    )

    #
    =====
    # FILE SIZE

```

```

#
=====

import os

keras_size = os.path.getsize(
    'ml/model/dnn_final_realistic.keras'
)/1024

tflite_size = os.path.getsize(
    'ml/model/dnn_final_realistic.tflite'
)/1024

print("\n=====")
print(" MODEL SIZE ")
print("=====")

print(
f"Keras : {keras_size:.2f} KB"
)

print(
f"TF Lite : {tflite_size:.2f} KB"
)

```