

LAMPIRAN

Lampiran 1 Surat Kesiediaan Membimbing TA

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 : Rifqi Naufal Aziz
NIM : 23040078
Program Studi : Diploma III Teknik Komputer
Judul TA : IMPLEMENTASI LOGISTIC REGRESSION DENGAN
POLYNOMIAL FEATURES UNTUK KLASIFIKASI
KONDISI PIPA BERDASARKAN DEBIT DAN
TEKANAN AIR

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana semestinya.

Tegal, 10 Juni 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 Kesiediaan Membimbing TA

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 : Rifqi Naufal Aziz
NIM : 23040078
Program Studi : Diploma III Teknik Komputer
Judul TA : IMPLEMENTASI LOGISTIC REGRESSION DENGAN
POLYNOMIAL FEATURES UNTUK KLASIFIKASI
KONDISI PIPA BERDASARKAN DEBIT DAN
TEKANAN AIR

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana semestinya.

Tegal, 10 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



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. Dokumentasi



Lampiran 5. Bimbingan Pembimbing I

Laporan : Abdul Basit, S.Kom, MT.

No	HARI/ TANGGAL	URAIAN	TANDA TANGAN
1.	26 Mei 2026	- Revisi Bab III Metodologi Penelitian Alur penelitian Revisi Atas Fokus ML	
2.		Revisi Bab 12.13 dan	

Lampiran 6. Bimbingan Pembimbing II

Lampiran 23
Bimbingan Laporan Pembimbing I TA

IK P2M PHB d.5.1.e.1

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

No	HARI/ TANGGAL	URAIAN	TANDA TANGAN
1.	1 April 2026	- Revisi Judul - Mencoba Training menggunakan Linear Linear Regresi dan KNN - Model diubah agar Akurasi naik dari 81% ke 90%	h
2.	20 April 2026	- Fokus Machine Learning - Augmentasi Data - SMOTE - Coba algoritma ML	h
3.	12 Juni 2026	- Kalimat awal tidak boleh kata asing - Teori terkait - Tambahkan Jurnal - Daftar Gambar huruf pertama besar - Library bab 2 landasan teori	h
4.	15 Juni	- tidak boleh kata akan - tidak boleh kata dihapuskan - data baglee diceritakan - bunyikan dataset sudah terlabeli	h

No	HARI/ TANGGAL	URAIAN	TANDA TANGAN
	15 JUNI 2026	- Tambahkan satu kolom Tabel untuk menampilkan data - dataset Augmentasi: jumlah datanya berapa - kata berikut yang di kata petunjuk hapus - Hasil Gambar disediakan - Rumus Masukan Logistic Regresi - Gambar di frame	h
5	19 JUNI	- Polynomial features Masukan dan keluar data - Revisi frame 9 tabel - kalimat "Berdasarkan Gambar atau tabel" ini dihapus karena diatas tabel sudah ada kata petunjuk	h

No	HARI/ TANGGAL	URAIAN	TANDA TANGAN
		_____ Sip dirj. ben _____	h 26/ 6 - 2026

Lampiran 7. Source Code

```
from google.colab import drive
drive.mount('/content/drive')

DIR_DATASET = '/content/drive/MyDrive/Colab
Notebooks/dataset/'
DIR_GAMBAR = '/content/drive/MyDrive/Colab
Notebooks/gambar visualisasi/'
DIR_MODEL = '/content/drive/MyDrive/Colab
Notebooks/model/'
DIR_TESTING = '/content/drive/MyDrive/Colab
Notebooks/testing/'
DIR_VISUAL_TESTING = '/content/drive/MyDrive/Colab
Notebooks/gambar visual testing/'

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import joblib

from scipy import stats
from sklearn.model_selection import train_test_split,
GridSearchCV
from sklearn.preprocessing import PolynomialFeatures,
StandardScaler
from sklearn.linear_model import LogisticRegression
from imblearn.over_sampling import SMOTE
from imblearn.pipeline import Pipeline
from sklearn.metrics import (
    classification_report,
    confusion_matrix,
    accuracy_score,
    precision_recall_fscore_support
)
df_orig = pd.read_csv(DIR_DATASET +
"location_aware_gis_leakage_dataset.csv") [ ["Pressure",
"Flow_Rate", "Leakage_Flag"]]
df_aug1 = pd.read_csv(DIR_DATASET +
"augmentasi_1.csv") [ ["Pressure", "Flow_Rate",
"Leakage_Flag"]]
df_aug2 = pd.read_csv(DIR_DATASET +
"augmentasi_2.csv") [ ["Pressure", "Flow_Rate",
"Leakage_Flag"]]
df = pd.concat(
```

```

        [df_orig, df_aug1, df_aug2],
        ignore_index=True
    )
    print("=====")
    print(" INFORMASI DATASET AWAL")
    print("=====")
    print(f"Total data awal: {len(df)}")
    print("\nDistribusi label awal:")
    print(df["Leakage_Flag"].value_counts())
    #
    =====
    # VISUALISASI INFORMASI DATASET AWAL
    #
    =====
    label_awal =
    df["Leakage_Flag"].value_counts().sort_index()
    df_label_awal = pd.DataFrame({
        "Status": ["Normal", "Bocor"],
        "Jumlah": [
            label_awal.get(0, 0),
            label_awal.get(1, 0)
        ]
    })
    plt.figure(figsize=(6, 4))

    sns.barplot(
        data=df_label_awal,
        x="Status",
        y="Jumlah"
    )
    plt.title("Informasi Dataset Awal - Distribusi Label")
    plt.xlabel("Status Kebocoran")
    plt.ylabel("Jumlah Data")
    for index, row in df_label_awal.iterrows():
        plt.text(
            index,
            row["Jumlah"],
            f'{row["Jumlah"]:,}'.replace(",", "."),
            ha="center",
            va="bottom"
        )
    plt.tight_layout()
    plt.savefig(DIR_GAMBAR +
    "1_distribusi_dataset_awal.png", dpi=300)
    plt.show()
    #
    =====

```

```

# 2. PEMBERSIHAN DATA (PREPROCESSING)
#
=====
print("\n=====")
print(" INFORMASI JUMLAH DATA PREPROCESSING")
print("=====")
print(f>Data Awal                                : {len(df)}")
# 1. Menghapus data kosong
df = df.dropna()
print(f>Setelah Hapus Missing Value              : {len(df)}")
# 2. Menghapus data duplikat
df = df.drop_duplicates()
print(f>Setelah Hapus Data Duplikat              : {len(df)}")
# 3. Memastikan tipe data numerik
df["Pressure"] = pd.to_numeric(df["Pressure"],
errors="coerce")
df["Flow_Rate"] = pd.to_numeric(df["Flow_Rate"],
errors="coerce")
df["Leakage_Flag"] = pd.to_numeric(df["Leakage_Flag"],
errors="coerce")
df = df.dropna()
print(f>Setelah Konversi Numerik                  : {len(df)}")
# 4. Menghapus outlier menggunakan Z-Score
z_scores = np.abs(stats.zscore(df[["Pressure",
"Flow_Rate"]]))
df = df[(z_scores < 3).all(axis=1)]
print(f>Setelah Z-Score (Hapus Outlier)          : {len(df)}")
# 5. Smoothing menggunakan Moving Average
df["Pressure"] = df["Pressure"].rolling(window=3,
min_periods=1).mean()
df["Flow_Rate"] = df["Flow_Rate"].rolling(window=3,
min_periods=1).mean()
print(f>Setelah Moving Average                    : {len(df)}")
# 6. Cleaning ulang setelah smoothing
df = df.dropna().drop_duplicates()
print(f>Setelah Cleaning Ulang                    : {len(df)}")
print("\n=====")
print(" INFORMASI DATASET SETELAH CLEANING")
print("=====")
print(f>Total data setelah dibersihkan            : {len(df)}")
print("\nDistribusi label setelah cleaning:")
print(df["Leakage_Flag"].value_counts())
#
=====
# VISUALISASI INFORMASI DATASET SETELAH CLEANING
#
=====

```

```

label_cleaning
df["Leakage_Flag"].value_counts().sort_index()
total_cleaning = len(df)
data_normal = label_cleaning.get(0, 0)
data_bocor = label_cleaning.get(1, 0)

df_info_cleaning = pd.DataFrame({
    "Keterangan": [
        "Total Cleaning",
        "Data Normal",
        "Data Bocor"
    ],
    "Jumlah Data": [
        total_cleaning,
        data_normal,
        data_bocor
    ]
})

plt.figure(figsize=(8, 5))

sns.barplot(
    data=df_info_cleaning,
    x="Keterangan",
    y="Jumlah Data"
)

plt.title("Informasi Dataset Setelah Cleaning")
plt.xlabel("Keterangan")
plt.ylabel("Jumlah Data")

for index, row in df_info_cleaning.iterrows():
    plt.text(
        index,
        row["Jumlah Data"],
        f'{row["Jumlah Data"]:,}'.replace(",", " "),
        ha="center",
        va="bottom"
    )

plt.xticks(rotation=0)
plt.tight_layout()
plt.savefig(DIR_GAMBAR
"2_informasi_dataset_setelah_cleaning.png", dpi=300)
plt.show()

print("\nTabel Informasi Dataset Setelah Cleaning:")

```

```

print(df_info_cleaning.to_string(index=False))
#
=====
# 3. PEMISAHAN FITUR DAN TARGET
#
=====

X = df[["Pressure", "Flow_Rate"]]
y = df["Leakage_Flag"]

X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.2,
    random_state=42,
    stratify=y
)
print("\n=====")
print(" DATA TRAINING DAN TESTING")
print("=====")
print(f"Jumlah data training: {len(X_train)}")
print(f"Jumlah data testing : {len(X_test)}")
print("\nDistribusi label training:")
print(y_train.value_counts())
print("\nDistribusi label testing:")
print(y_test.value_counts())
#
=====
# VISUALISASI PEMBAGIAN DATA TRAINING DAN TESTING
#
=====
jumlah_data = pd.DataFrame({
    "Jenis Data": ["Training", "Testing"],
    "Jumlah": [len(X_train), len(X_test)]
})
plt.figure(figsize=(6, 4))
sns.barplot(
    data=jumlah_data,
    x="Jenis Data",
    y="Jumlah"
)
plt.title("Pembagian Data Training dan Testing")
plt.xlabel("Jenis Data")
plt.ylabel("Jumlah Data")

for index, row in jumlah_data.iterrows():
    plt.text(

```

```

        index,
        row["Jumlah"],
        f'{row["Jumlah"]:,}'.replace(",", ".") ,
        ha="center",
        va="bottom"
    )
plt.tight_layout()
plt.savefig(DIR_GAMBAR                                     +
"3_pembagian_data_training_testing.png", dpi=300)
plt.show()
#
=====
# VISUALISASI DISTRIBUSI LABEL DATA TRAINING
#
=====
label_training = y_train.value_counts().sort_index()
df_train_visual = pd.DataFrame({
    "Status": ["Normal", "Bocor"],
    "Jumlah": [
        label_training.get(0, 0),
        label_training.get(1, 0)
    ]
})

plt.figure(figsize=(6, 4))

sns.barplot(
    data=df_train_visual,
    x="Status",
    y="Jumlah"
)
plt.title("Distribusi Label Data Training")
plt.xlabel("Status Kebocoran")
plt.ylabel("Jumlah Data")
for index, row in df_train_visual.iterrows():
    plt.text(
        index,
        row["Jumlah"],
        f'{row["Jumlah"]:,}'.replace(",", ".") ,
        ha="center",
        va="bottom"
    )
plt.tight_layout()
plt.savefig(DIR_GAMBAR                                     +
"4_distribusi_label_training.png", dpi=300)
plt.show()

```

```

#
=====
# VISUALISASI DISTRIBUSI LABEL DATA TESTING
#
=====
label_testing = y_test.value_counts().sort_index()
df_test_visual = pd.DataFrame({
    "Status": ["Normal", "Bocor"],
    "Jumlah": [
        label_testing.get(0, 0),
        label_testing.get(1, 0)
    ]
})

plt.figure(figsize=(6, 4))
sns.barplot(
    data=df_test_visual,
    x="Status",
    y="Jumlah"
)
plt.title("Distribusi Label Data Testing")
plt.xlabel("Status Kebocoran")
plt.ylabel("Jumlah Data")

for index, row in df_test_visual.iterrows():
    plt.text(
        index,
        row["Jumlah"],
        f'{row["Jumlah"]:,}'.replace(",", "."),
        ha="center",
        va="bottom"
    )
plt.tight_layout()
plt.savefig(DIR_GAMBAR +
"5_distribusi_label_testing.png", dpi=300)
plt.show()
#
=====
# 4. MEMBUAT MODEL PIPELINE
#
=====
model_lr_poly = Pipeline([
    ("poly", PolynomialFeatures()),
    ("scaler", StandardScaler()),
    ("smote", SMOTE(random_state=42)),
    ("model", LogisticRegression(random_state=42))
])

```

```

#
=====
# 5. PARAMETER GRIDSEARCHCV
#
=====
param_grid = {
    "poly__degree": [2, 3],
    "model__C": [0.01, 0.1, 1, 10, 100],
    "model__solver": ["lbfgs", "liblinear"],
    "model__max_iter": [1000, 2000, 3000]
}
#
=====
# 6. TRAINING DENGAN GRIDSEARCHCV
#
=====
grid_search = GridSearchCV(
    estimator=model_lr_poly,
    param_grid=param_grid,
    cv=5,
    scoring="accuracy",
    n_jobs=-1
)
print("\n=====")
print(" PROSES TRAINING")
print("=====")

grid_search.fit(X_train, y_train)
print("\n[INFO] Training selesai.")

# Mengambil model terbaik
best_model = grid_search.best_estimator_
print("\n=====")
print(" HASIL PARAMETER TERBAIK")
print("=====")
print(grid_search.best_params_)
print(f"Akurasi Cross Validation terbaik:
{grid_search.best_score_ * 100:.2f}%")
#
=====
# 7. EVALUASI MODEL
#
=====
y_pred = best_model.predict(X_test)
acc = accuracy_score(y_test, y_pred)
print("\n=====")
print(" HASIL EVALUASI MODEL")

```

```

print("=====")
print(f"Accuracy Training: {acc * 100:.2f}%")
print("\nClassification Report:")
print(classification_report(
    y_test,
    y_pred,
    labels=[0, 1],
    target_names=["Normal", "Bocor"]
))
#
=====
# VISUALISASI HASIL EVALUASI MODEL
#
=====
precision,      recall,      f1_score,      support      =
precision_recall_fscore_support(
    y_test,
    y_pred,
    labels=[0, 1]
)

kelas = ["Normal", "Bocor"]

x = np.arange(len(kelas))
width = 0.25

plt.figure(figsize=(8, 5))

# Warna diagram diganti
bar1 = plt.bar(x - width, precision, width,
label="Precision", color="#6A5ACD")
bar2 = plt.bar(x, recall, width, label="Recall",
color="#20B2AA")
bar3 = plt.bar(x + width, f1_score, width, label="F1-
Score", color="#FFB347")
# Menampilkan nilai di atas batang
for bars in [bar1, bar2, bar3]:
    for bar in bars:
        height = bar.get_height()
        plt.text(
            bar.get_x() + bar.get_width() / 2,
            height + 0.01,
            f"{height:.2f}",
            ha="center",
            va="bottom",
            fontsize=9
        )

```

```

plt.title("Hasil Evaluasi Model Logistic Regression
(Pengujian Internal)")
plt.xlabel("Kelas")
plt.ylabel("Nilai")
plt.xticks(x, kelas)
plt.ylim(0, 1.1)
plt.legend()
plt.grid(axis="y", linestyle="--", alpha=0.4)

plt.tight_layout()
plt.savefig(DIR_GAMBAR + "6_hasil_evaluasi_model.png",
dpi=300)
plt.show()
#
=====
# 8. CONFUSION MATRIX
#
=====
==
cm = confusion_matrix(y_test, y_pred, labels=[0, 1])
print("\nConfusion Matrix:")
print(cm)

print("\nKeterangan Confusion Matrix:")
print("TN / Normal diprediksi Normal :", cm[0][0])
print("FP / Normal diprediksi Bocor :", cm[0][1])
print("FN / Bocor diprediksi Normal :", cm[1][0])
print("TP / Bocor diprediksi Bocor :", cm[1][1])

plt.figure(figsize=(6, 5))
sns.heatmap(
    cm,
    annot=True,
    fmt="d",
    cmap="Greens",
    xticklabels=["Normal", "Bocor"],
    yticklabels=["Normal", "Bocor"]
)

plt.title("Confusion Matrix")
plt.xlabel("Prediksi")
plt.ylabel("Aktual")

plt.tight_layout()
plt.savefig(DIR_GAMBAR + "7_confusion_matrix.png",
dpi=300)

```

```

plt.show()
#
=====
# 9. SIMPAN MODEL
#
=====
path_simpan_model = DIR_MODEL + "logisticregresion.pkl"
joblib.dump(best_model, path_simpan_model)
print("\n=====")
print(" MODEL BERHASIL DISIMPAN DI GOOGLE DRIVE")
print("=====")
print(f"File: {path_simpan_model}")
#
=====
# 10. SIMPAN DATA CSV HASIL TRAINING
#
=====
# Menggabungkan kembali X_train dan y_train menjadi satu
DataFrame
df_training = pd.concat([X_train, y_train], axis=1)

# Menyimpan ke penyimpanan lokal Colab (bukan Google
Drive)
path_simpan_csv = "data_training_hasil.csv"
df_training.to_csv(path_simpan_csv, index=False)

print("\n=====")
print(" DATA TRAINING BERHASIL DISIMPAN DI COLAB")
print("=====")
print(f"File          CSV          tersimpan          di:
/content/{path_simpan_csv}")

```