

# LAMPIRAN

## Lampiran A. Data Latih Sensor

```
File Edit Selection Find View Goto Tools Project Preferences Help
train.py — C:\Users\ASUS  run1.py  run.py  run.py

1 rms,db,zcr,class
2 0.005732,75.17,0.521000,dentuman
3 0.008033,78.10,0.537000,dentuman
4 0.008457,78.54,0.518000,dentuman
5 0.012292,81.79,0.545000,dentuman
6 0.013669,82.71,0.576000,dentuman
7 0.014774,83.39,0.547000,dentuman
8 0.013675,82.72,0.573000,dentuman
9 0.013843,82.82,0.560000,dentuman
10 0.012956,82.25,0.545000,dentuman
11 0.013196,82.41,0.547000,dentuman
12 0.014299,83.11,0.557000,dentuman
13 0.012555,81.98,0.583000,dentuman
14 0.013526,82.62,0.555000,dentuman
15 0.012948,82.24,0.558000,dentuman
16 0.012243,81.76,0.540000,dentuman
17 0.013433,82.56,0.566000,dentuman
18 0.012060,81.63,0.580000,dentuman
19 0.014816,83.41,0.560000,dentuman
20 0.010991,80.82,0.519000,dentuman
21 0.014219,83.06,0.557000,dentuman
22 0.009625,79.67,0.519000,dentuman
23 0.012976,82.26,0.557000,dentuman
24 0.013163,82.39,0.555000,dentuman
25 0.013765,82.78,0.553000,dentuman
26 0.013430,82.56,0.547000,dentuman
27 0.014854,83.44,0.572000,dentuman
28 0.013085,82.34,0.548000,dentuman
29 0.013440,82.57,0.566000,dentuman
30 0.014426,83.18,0.551000,dentuman
31 0.013593,82.67,0.535000,dentuman
32 0.012817,82.16,0.548000,dentuman
33 0.015984,84.07,0.552000,dentuman
34 0.000871,58.80,0.056000,manusia
35 0.000742,57.41,0.062000,manusia
36 0.001513,63.60,0.067000,manusia
37 0.001327,62.46,0.074000,manusia
38 0.001055,60.47,0.057000,manusia
39 0.000949,59.55,0.065000,manusia
40 0.000892,59.01,0.054000,manusia

Line 1, Column 1

File Edit Selection Find View Goto Tools Project Preferences Help
train.py — C:\Users\ASUS  run1.py  run.py  ru

36 0.001513,63.60,0.067000,manusia
37 0.001327,62.46,0.074000,manusia
38 0.001055,60.47,0.057000,manusia
39 0.000949,59.55,0.065000,manusia
40 0.000892,59.01,0.054000,manusia
41 0.001200,61.59,0.239000,manusia
42 0.001013,60.11,0.059000,manusia
43 0.000890,58.99,0.057000,manusia
44 0.000853,58.61,0.052000,manusia
45 0.000882,58.90,0.046000,manusia
46 0.001244,61.90,0.068000,manusia
47 0.001141,61.14,0.064000,manusia
48 0.000922,59.30,0.052000,manusia
49 0.000711,57.04,0.066000,manusia
50 0.000848,58.57,0.064000,manusia
51 0.001614,64.16,0.060000,manusia
52 0.002071,66.33,0.064000,manusia
53 0.000852,58.61,0.079000,manusia
54 0.003542,70.98,0.049000,manusia
55 0.001173,61.38,0.062000,manusia
56 0.001008,60.07,0.066000,manusia
57 0.001485,63.43,0.072000,manusia
58 0.001266,62.05,0.077000,manusia
59 0.002067,66.31,0.042000,manusia
60 0.001140,61.14,0.095000,manusia
61 0.000658,56.37,0.066000,manusia
62 0.000712,57.05,0.071000,manusia
63 0.000749,57.49,0.071000,manusia
64 0.000885,58.94,0.059000,manusia
65 0.001171,61.37,0.078000,manusia
66 0.001057,60.48,0.064000,manusia
67 0.001211,61.66,0.062000,manusia
68 0.001239,61.86,0.051000,manusia
69 0.001528,63.68,0.055000,manusia
70 0.000734,57.31,0.054000,manusia
71 0.001131,61.07,0.074000,manusia
72 0.000902,59.11,0.090000,manusia
73 0.000644,56.18,0.072000,manusia
74 0.000871,58.80,0.068000,manusia
75 0.002349,67.42,0.060000,manusia

Line 1, Column 1
```

```
File Edit Selection Find View Goto Tools Project Preferences H
train.py — C:\Users\ASUS | run1.py | run.py | run
337 0.007003,76.91,0.214000,musik
338 0.006204,75.85,0.273000,musik
339 0.008458,78.55,0.476000,musik
340 0.008497,78.58,0.338000,musik
341 0.010218,80.19,0.441000,musik
342 0.009044,79.13,0.268000,musik
343 0.009683,79.72,0.277000,musik
344 0.008691,78.78,0.354000,musik
345 0.012402,81.87,0.303000,musik
346 0.007938,77.99,0.275000,musik
347 0.008413,78.50,0.349000,musik
348 0.007340,77.31,0.267000,musik
349 0.006791,76.64,0.242000,musik
350 0.005968,75.52,0.348000,musik
351 0.008240,78.32,0.289000,musik
352 0.007032,76.94,0.324000,musik
353 0.006494,76.25,0.337000,musik
354 0.009287,79.36,0.377000,musik
355 0.005773,75.23,0.301000,musik
356 0.007664,77.69,0.430000,musik
357 0.002708,68.65,0.165000,musik
358 0.005983,75.54,0.225000,musik
359 0.002001,66.03,0.173000,musik
360 0.001956,65.83,0.142000,musik
361 0.002690,68.60,0.189000,musik
362 0.002009,66.06,0.129000,musik
363 0.002068,66.31,0.184000,musik
364 0.001572,63.93,0.158000,musik
365 0.003015,69.58,0.211000,musik
366 0.001379,62.79,0.238000,musik
367 0.001978,65.92,0.219000,musik
368 0.005763,75.21,0.316000,musik
369 0.002516,68.01,0.183000,musik
370 0.003395,70.62,0.241000,musik
371 0.004985,73.95,0.281000,musik
372 0.001798,65.10,0.233000,musik
373 0.002665,68.52,0.179000,musik
374 0.001860,65.39,0.140000,musik
375 0.002692,68.60,0.300000,musik
376 0.002008,66.05,0.077000,musik
Line 1, Column 1
```

## Lapiran B. Script Data Train Python

```

File Edit Selection Find View Goto Tools Project Preferences Help
train.py - C:\Users\ASUS > run1.py > runpy > runnpy > train1.py x
1 import pandas as pd
2 from sklearn.preprocessing import StandardScaler
3 from sklearn.neighbors import KNeighborsClassifier
4 from sklearn.model_selection import train_test_split
5 from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
6 import joblib
7 import matplotlib.pyplot as plt
8 import seaborn as sns
9
10 plt.style.use("default")
11
12 # =====
13 # 1. LOAD DATASET
14 # =====
15 df = pd.read_csv("data_latih.txt") # bisa .csv/.txt
16
17 print("\n== 5 Data Portama ==")
18 print(df.head())
19
20 # Ambil fitur & label
21 X = df[["rms", "db", "zcr"]]
22 y = df["class"]
23
24 # =====
25 # 2. NORMALISASI FITUR
26 # =====
27 scaler = StandardScaler()
28 X_scaled = scaler.fit_transform(X)
29
30 # =====
31 # 3. SPLIT DATA
32 # =====
33 X_train, X_test, y_train, y_test = train_test_split(
34     X_scaled, y, test_size=0.2, random_state=42, stratify=y
35 )
36
37 # =====
38 # 4. TRAIN MODEL KNN
39 # =====
40 model = KNeighborsClassifier(
41     n_neighbors=5,
42     weights="distance",
43     metrics="minkowski",
44     p=2
45 )
46
47 model.fit(X_train, y_train)
48
49 # =====
50 # 5. PREDIKSI & EVALUASI
51 # =====
52 y_pred = model.predict(X_test)
53 acc = accuracy_score(y_test, y_pred)
54 cm = confusion_matrix(y_test, y_pred)
55
56 print("\n== HASIL EVALUASI ==")
57 print(f"Akurasi Model: {acc*100:.2f}%\n")
58
59 print("Classification Report:")
60 print(classification_report(y_test, y_pred))
61
62 print("Confusion Matrix:")
63 print(cm)
64
65 # =====
66 # 6. VISUALISASI CONFUSION MATRIX
67 # =====
68 plt.figure(figsize=(6,5))
69 sns.heatmap(cm, annot=True, cmap="Blues", fmt="d",
70             xticklabels=model.classes_,
71             yticklabels=model.classes_)
72
73 plt.title("Confusion Matrix KNN (Deteksi Kebisingan)")
74 plt.xlabel("Prediksi")
75 plt.ylabel("Aktual")
76 plt.tight_layout()
77 plt.show()
78
79 # =====
80 # 7. SIMPAN MODEL & SCALER
81 # =====
82 joblib.dump(model, "knn_model.pkl")
83 joblib.dump(scaler, "scaler.pkl")
84
85 print("\nModel & scaler berhasil disimpan:")
86 print(" -> knn_model.pkl")
87 print(" -> scaler.pkl")

```

```

File Edit Selection Find View Goto Tools Project Preferences Help
train.py - C:\Users\ASUS > run1.py > runpy > runnpy > train1.py x
56 print("\n== HASIL EVALUASI ==")
57 print(f"Akurasi Model: {acc*100:.2f}%\n")
58
59 print("Classification Report:")
60 print(classification_report(y_test, y_pred))
61
62 print("Confusion Matrix:")
63 print(cm)
64
65 # =====
66 # 6. VISUALISASI CONFUSION MATRIX
67 # =====
68 plt.figure(figsize=(6,5))
69 sns.heatmap(cm, annot=True, cmap="Blues", fmt="d",
70             xticklabels=model.classes_,
71             yticklabels=model.classes_)
72
73 plt.title("Confusion Matrix KNN (Deteksi Kebisingan)")
74 plt.xlabel("Prediksi")
75 plt.ylabel("Aktual")
76 plt.tight_layout()
77 plt.show()
78
79 # =====
80 # 7. SIMPAN MODEL & SCALER
81 # =====
82 joblib.dump(model, "knn_model.pkl")
83 joblib.dump(scaler, "scaler.pkl")
84
85 print("\nModel & scaler berhasil disimpan:")
86 print(" -> knn_model.pkl")
87 print(" -> scaler.pkl")

```

## Lampiran C. Script Realtime Predict Python.

```
File Edit Selection Find View Goto Tools Project Preferences Help
train1.py x Train0.py x run2.py
37
40 # -----
41 # FASTAPI APP
42 # -----
43 app = FastAPI(title="Noise-Classifer Receiver")
44
45 # -----
46 # Load model & scaler
47 # -----
48 model = None
49 scaler = None
50
51 if os.path.exists(MODEL_FILE):
52     try:
53         model = joblib.load(MODEL_FILE)
54         print(f"[MODEL] Loaded {MODEL_FILE}")
55     except:
56         print("[MODEL] Failed to load model")
57 else:
58     print("[MODEL] Model not found")
59
60 if os.path.exists(SCALER_FILE):
61     try:
62         scaler = joblib.load(SCALER_FILE)
63         print(f"[SCALER] Loaded {SCALER_FILE}")
64     except:
65         print("[SCALER] Failed to load scaler")
66 else:
67     print("[SCALER] Scaler not found")
68
69
70 # -----
71 # Pydantic model
72 # -----
73 class Sample(BaseModel):
74     node: str | int = "unknown"
75     rms: Decimal
76     db: Decimal | None = None
77     zcr: Decimal
78     spl: Decimal | None = None
Line 60, Column 23
```

```
File Edit Selection Find View Goto Tools Project Preferences Help
train1.py x Train0.py x run2.py
1 # run.py
2 from fastapi import FastAPI, Request
3 from pydantic import BaseModel
4 import uvicorn
5 import joblib
6 import datetime
7 import requests
8 import os
9 import traceback
10 import numpy as np
11 from decimal import Decimal
12 import warnings
13
14 # -----
15 # HILANGKAN WARNING SKLEARN
16 # -----
17 warnings.filterwarnings('ignore', message='X does not have valid feature names')
18
19 # -----
20 # CONFIG
21 # -----
22 MODEL_FILE = "km_model.pkl"
23 SCALER_FILE = "scaler.pkl"
24
25 HOST = "0.0.0.0"
26 PORT = 5000
27
28 LOG_FILE = "log_data.txt"
29 LOG_DATASET = "dataset_log.txt" # <--- file dataset training
30
31 # Threshold untuk alert
32 THRESHOLD_DB = 70.0
33 THRESHOLD_RMS = 0.6
34 THRESHOLD_ZCR = 0.9
35
36 # Telegram
37 TELEGRAM_TOKEN = "8008220335:AAW51ry300HwHwM2x251o69X_t6L1f0Hg"
38 CHAT_ID = "1478583665"
39
40 # -----
Line 60, Column 23
```

```
File Edit Selection Find View Goto Tools Project Preferences Help
train1.py x Train0.py x run2.py
77
78 zcr: Decimal
79 spl: Decimal | None = None
80
81 # -----
82 # Helper: save raw log (debug)
83 # -----
84 def save_log(text):
85     with open(LOG_FILE, "a", encoding="utf-8") as f:
86         f.write(text + "\n")
87
88
89 # -----
90 # Helper: save dataset rows (training file)
91 # -----
92 def save_dataset_row(rms, db, zcr, pred):
93     """
94     Simpan dataset dalam format:
95     rms,db,zcr,class
96     Digunakan untuk training.
97     """
98     write_header = not os.path.exists(LOG_DATASET)
99
100     with open(LOG_DATASET, "a", encoding="utf-8") as f:
101         if write_header:
102             f.write("rms,db,zcr,class\n")
103         f.write(f"{rms:.6f},{db:.2f},{zcr:.6f},{pred}\n")
104
105
106 # -----
107 # Helper: Telegram
108 # -----
109
110 def send_telegram_message(text):
111     if not TELEGRAM_TOKEN or not CHAT_ID:
112         print("[TELEGRAM] SKIP (token/chat_id kosong)")
113         return
114     url = f"https://api.telegram.org/bot{TELEGRAM_TOKEN}/sendMessage"
115     requests.post(url, data={"chat_id": CHAT_ID, "text": text})
Line 60, Column 23
```

```
File Edit Selection Find View Goto Tools Project Preferences Help
train1.py x Train0.py x run2.py
155     "alert": False,
156     "alert_reasons": []
157 }
158
159
160 try:
161     # --- Feature untuk model ---
162     feature = np.array([[rms, db, zcr]])
163
164     if scaler:
165         try:
166             feature = scaler.transform(feature)
167         except:
168             pass
169
170     # --- Predict ---
171     if model:
172         pred = model.predict(feature)[0]
173         result["prediction"] = str(pred)
174
175         # SIMPAN DATASET UNTUK TRAINING
176         save_dataset_row(rms, db, zcr, pred)
177
178     else:
179         pred = "model_not_loaded"
180         result["prediction"] = pred
181
182     # --- Threshold alert ---
183     if db >= THRESHOLD_DB:
184         result["alert"] = True
185         result["alert_reasons"].append("db >= threshold")
186
187     if rms >= THRESHOLD_RMS:
188         result["alert"] = True
189         result["alert_reasons"].append("rms >= threshold")
190
191     if zcr >= THRESHOLD_ZCR:
192         result["alert"] = True
193         result["alert_reasons"].append("zcr >= threshold")
194
195     # ---
Line 60, Column 23
```

```
File Edit Selection Find View Goto Tools Project Preferences Help
train1.py x Train0.py x run2.py
116 requests.post(url, data={"chat_id": CHAT_ID, "text": text})
117
118
119 # -----
120 # API: PREDICT
121 # -----
122 @app.post("/api/predict")
123 async def api_predict(sample: Sample, request: Request):
124
125     now = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
126     node = str(sample.node)
127
128     # -----
129     # FIX PRECISI FLOAT
130     # -----
131     rms = float(sample.rms)
132     zcr = float(sample.zcr)
133     db = float(sample.db) if sample.db is not None else (
134         float(sample.spl) if sample.spl is not None else 0.0
135     )
136
137     # Format string untuk log
138     rms_str = f"{rms:.6f}"
139     zcr_str = f"{zcr:.6f}"
140     db_str = f"{db:.2f}"
141
142     # Debug print
143     print(f"[RECV] node={node}, rms={rms_str}, db={db_str}, zcr={zcr_str}")
144
145     # Raw log
146     save_log(f"{now},node={node},rms={rms_str},db={db_str},zcr={zcr_str}")
147
148     result = {
149         "node": node,
150         "rms": rms,
151         "db": db,
152         "zcr": zcr,
153         "timestamp": now,
154         "prediction": None,
155         "alert": False
156     }
157     return result
158
159 if __name__ == '__main__':
160     app.run(host='0.0.0.0', port=5000)
```

Line 60, Column 23



## Lampiran E. Script ESP Pusat

```

sketch_node1.ino
119
120 float rms = 0, spl = 0, zcr = 0;
121 computeFeatures(rms, spl, zcr, valid_samples);
122
123 Serial.printf("RMS: %.6f | SPL: %.2f dB | ZCR: %.3f\n", rms, spl, zcr);
124
125 // ===== SEND JSON =====
126 String json = "{";
127 json += "\"node\": \"node1\", ";
128 json += "\"rms\": " + String(rms, 6) + ", ";
129 json += "\"spl\": " + String(spl, 2) + ", ";
130 json += "\"zcr\": " + String(zcr, 3);
131 json += "}";
132
133 // ===== Kirim ke ESP Pusat =====
134 if (WiFi.status() == WL_CONNECTED) {
135   HTTPClient http;
136   String url = "http://" + PUSAT_IP + endpoint;
137
138   http.begin(url);
139   http.addHeader("Content-Type", "application/json");
140
141   int code = http.POST(json);
142   Serial.print("Send + ");
143   Serial.println(code);
144
145   http.end();
146
147   delay(500);
148 }
149
150
151
sketch_esp_pusat.ino
1
#include <WiFi.h>
2
#include <WebServer.h>
3
#include <ArduinoJson.h>
4
#include <Wire.h>
5
#include <HTTPClient.h>
6
#include <LiquidCrystal_I2C.h>
7
LiquidCrystal_I2C lcd(0x27, 20, 4);
8
const char* ssid = "MI 12";
9
const char* password = "12345677";
10
11
12
WebServer server(80);
13
14
// ===== DATA NODE =====
15
struct NodeData {
16   float rms = 0;
17   float db = 0;
18   float zcr = 0;
19   unsigned long lastSeen = 0;
20   bool active = false;
21 };
22
NodeData node1;
24
NodeData node2;
25
26
27 // hasil prediksi dari backend
28 String classNode1 = "-";
29 String classNode2 = "-";
30
31 // ===== SEND TO PYTHON =====
32 void sendToPython(float rms, float db, float zcr, int nodeID) {
33   WiFiClient client;
34   Serial Monitor X
35
sketch_esp_pusat.ino
33 WiFiClient client;
34 HTTPClient http;
35
36 http.begin(client, "http://10.229.164.141:5000/api/predict");
37 http.addHeader("Content-Type", "application/json");
38
39 // ----- FLEX PRESISI -----
40 String json = "{";
41 json += "\"node\": " + String(nodeID) + ", ";
42 json += "\"rms\": " + String(rms, 6) + ", "; // <<< FIX
43 json += "\"db\": " + String(db, 2) + ", "; // <<< FIX
44 json += "\"zcr\": " + String(zcr, 6); // <<< FIX
45 json += "}";
46
47 int code = http.POST(json);
48 Serial.print("[PUSAT + PYTHON] Status: ");
49 Serial.println(code);
50
51 if (code == 200) {
52   String payload = http.getString();
53   Serial.println("[PYTHON + RESP] " + payload);
54
55   StaticJsonDocument<256> doc;
56   if (!deserializeJson(doc, payload)) {
57     String pred = doc["prediction"].as<String>();
58
59     if (nodeID == 1) classNode1 = pred;
60     if (nodeID == 2) classNode2 = pred;
61   }
62 }
63
64 http.end();
65
sketch_esp_pusat.ino
64 http.end();
65 }
66
67 // ===== PARSE JSON =====
68 void parseNode(String body, NodeData &node) {
69   StaticJsonDocument<256> doc;
70
71   if (!deserializeJson(doc, body)) {
72     Serial.println("JSON ERROR");
73     return;
74   }
75
76   node.rms = doc["rms"].as<float>();
77
78   if (doc.containsKey("db"))
79     node.db = doc["db"].as<float>();
80   else if (doc.containsKey("spl"))
81     node.db = doc["spl"].as<float>();
82   else
83     node.db = 0.0;
84
85   node.zcr = doc["zcr"].as<float>();
86
87   node.lastSeen = millis();
88   node.active = true;
89 }
90
91 void handleNode1() {
92   if (!server.hasArg("plain")) return;
93
94   String body = server.arg("plain");
95   Serial.println("\n[DATA NODE 1]");
96   Serial.println(body);
97 }

```

```

sketch_esp_pusat.ino
96     Serial.println(body);
97
98     parseNode(body, node1);
99     sendToPython(node1.rms, node1.db, node1.zcr, 1);
100
101     server.send(200, "text/plain", "OK NODE 1");
102 }
103
104 void handleNode2() {
105     if (!server.hasArg("plain")) return;
106
107     String body = server.arg("plain");
108     Serial.println("\n[DATA NODE 2]");
109     Serial.println(body);
110
111     parseNode(body, node2);
112     sendToPython(node2.rms, node2.db, node2.zcr, 2);
113
114     server.send(200, "text/plain", "OK NODE 2");
115 }
116
117 unsigned long lastLCD = 0;
118
119 void updateLCD() {
120     if (millis() - lastLCD < 1000) return;
121     lastLCD = millis();
122
123     lcd.clear();
124
125     lcd.setCursor(0, 0);
126     lcd.print("N1:");
127     lcd.print(classNode1);
128     lcd.print(" ");
129     lcd.print(node1.db, 0);
130     lcd.print("dB");
131
132     lcd.setCursor(0, 1);
133     lcd.print("N2:");
134     lcd.print(classNode2);
135     lcd.print(" ");
136     lcd.print(node2.db, 0);
137     lcd.print("dB");
138
139     lcd.setCursor(0, 3);
140     lcd.print("IP:");
141     lcd.print(WiFi.localIP());
142 }
143
144 void setup() {
145     Serial.begin(115200);
146
147     lcd.init();
148     lcd.backlight();
149     lcd.print("ESP32 PUSAT...");
150
151     WiFi.begin(ssid, password);
152     Serial.print("Connecting");
153
154     while (WiFi.status() != WL_CONNECTED) {
155         delay(300);
156         Serial.print(".");
157     }
158
159     Serial.println("\nCONNECTED");
160     Serial.println(WiFi.localIP());
161
162     lcd.clear();
163     lcd.setCursor(0, 0); lcd.print("WiFi OK");
164     lcd.setCursor(0, 1); lcd.print(WiFi.localIP().toString());
165
166     server.on("/node1", HTTP_POST, handleNode1);
167     server.on("/node2", HTTP_POST, handleNode2);
168
169     server.begin();
170     Serial.println("SERVER RUNNING...");
171 }
172
173 void loop() {
174     server.handleClient();
175     updateLCD();
176 }
177

```

```

sketch_esp_pusat.ino
145     Serial.begin(115200);
146
147     lcd.init();
148     lcd.backlight();
149     lcd.print("ESP32 PUSAT...");
150
151     WiFi.begin(ssid, password);
152     Serial.print("Connecting");
153
154     while (WiFi.status() != WL_CONNECTED) {
155         delay(300);
156         Serial.print(".");
157     }
158
159     Serial.println("\nCONNECTED");
160     Serial.println(WiFi.localIP());
161
162     lcd.clear();
163     lcd.setCursor(0, 0); lcd.print("WiFi OK");
164     lcd.setCursor(0, 1); lcd.print(WiFi.localIP().toString());
165
166     server.on("/node1", HTTP_POST, handleNode1);
167     server.on("/node2", HTTP_POST, handleNode2);
168
169     server.begin();
170     Serial.println("SERVER RUNNING...");
171 }
172
173 void loop() {
174     server.handleClient();
175     updateLCD();
176 }
177

```

Output Serial Monitor X

Lampiran F. Dokumentasi Pengambilan Data Lobby 1



Lampiran G. Dokumentasi Pengambilan Data Lobby 2

