DAFTAR PUSTAKA

Nur Ikhsan Rahmani, Lelono Danang, Triyana Kuwat, Classification of cocoa based on e-nose with methods neuro fuzzy, Indonesian Journal of Electronics and Instrumentation Systems (IJEIS), Vol.8, No.1, April 2018.

Romantika Denting Java, Wijaya Wahyu, Hendry Jans, Sumanto Budi, Feature selection of e-nose system performance for gayo coffee aroma classification, Journal of Management, Informatics Engineering, and Computer Engineering Vol. 21, No. 2, March 2022.

Wafiyah Fakihatin, Hidayat Nurul, Perdana Setya Rizal, Implementation of the modified k-nearest neighbor (mknn) algorithm for classification of fever, Journal of Information Technology Development and Computer Science Vol. 1, No. 10, October 2017.

Yulia Gina Sari, Wildian, Firmawati Nini. Electronic nose (e-nose) system design for breast cancer sample detection microcontroller based on Arduino uno, Journal Of Physical Sciences (Jif), Vol 10 No 1, March 2018.

Khadafi Muammar, Almi Rachmad Putra, 2020, Electronic nose design (e-nose) for arduino nano-based halal haram identification, Journal of Physics and Its Applications, Vol. 13, No.1, Oktober 2020.

Taufan Maulana Putera, Dwiki Akmal Murad. 2017. GACOR (Gastric Acid Detector) : Inovasi Alat Pengukur Tingkat Asam Lambung. AJIE - Asian Journal of Innovation and Entrepreneurship (e-ISSN: 2477-0574 ; p-ISSN: 2477-3824) Vol. 06 , 01 January 2022.

KFH Hintzen , J Grote , AGWE Wintjens , T Lubbers ,corresponding author MMM Eussen , FJ van Schooten , ND Bouvy, 2021, Breath analysis for the detection of digestive tract malignancies: systematic review, Journal of PubMed Central, Vol. 5, April 2021.

Erik J. Snider, Griselda Compres, Daniel E. Freedberg, Marla J. Giddins, Hossein Khiabanian, Charles J. Lightdale, Yael R. Nobel, Nora C. Toussaint, Anne-Catrin Uhlemann, and Julian A. Abrams, 2018, Barrett's esophagus is associated with a distinct oral microbiome, Journal of PubMed Central, Vol. 9, Maret 2018.

Sergius Melashchenko, 2020, Assessment of gastric acidity by short-duration intragastric pH-monitoring with standardised breakfast in functional and some other dyspepsias, Journal of PubMed Central, Vol. 15, September 2020.

Rouhillah, Hasbiullah Ilham, Furqan Ahmad, Identification of fish with information gas sensor based, Jurnal J-Innovation Vol. 6, No. 2, Desember 2017.

Hariyanto, 2017, Non-invasive multilevel diabetes detection by analyzing human breath using a breathalyzer.

Yakob Muhammad, Mustika Dona, Miranda, Sari Nirmala, Putra Almi Rachmad. Design and development of an electronic nose (e-nose) to detect gastric disease (gastrointestinal gastritis and dyspepsia) through the respiratory tract, ceedings of the 2nd International Conference on Science, Technology and dern Society (ICSTMS 2020).

Dodi Putra, Alfita Riza, Rahmawati Diana, Design of a diabetes mellitus ection tool using backpropagation, Retrieved 21 January 2023.

Ozil Putra, Firdaus, Hafiz Mohammad Hersyah, Identification of tea

aromas with e-nose using the backpropagation method.
Aji Chrisal Lintang, Wahyu Triyogatama Widodo, Lelono Danang, Electronic nose design to detect freshwater fish decomposition rate.
Parvin Hamid, Alizadeh Hosein, Minaei Behrouz. 2008, MKNN: Modified k-nearest neighbor.
Siti Khadijah. 2019. Karakteristik penderita gerd. http://repository.unhas.ac.id/id/eprint/1200/2/C011171318_skripsi(FILEminimizer)_1-2.pdf. Diakses 9 Oktober 2021.
Arif, F. 2016. Mengenal Bagian-bagian Tampilan Aplikasi Arduino IDE. Https://www.pusatkomponen.com/blog/mengenal-tampilan-bagian-arduino-ide. Diakses 10 Oktober 2021.
Bagus. 2017. Bagian-bagian Utama Papan *Arduino Uno*. Http://webagus.id/index.php/2017/07/26/deskripsi-papan-arduino. Diakses 11 Oktober 2021.
Febriani. 2012. Definisi Perancangan. Http://febriani.staff.gunadarma.ac.id. Diakses 15 Oktober 2021.
Haryanto, T. 2016. *Analog Output* pada *Arduino* Menggunakan PWM *(Pulse Width Modulation)*. Https://www.codepolitan.com/analog-output-arduino-menggunakan-pwm-pulse-width-modulation. Diakses 15 Oktober 2021.
Rustamaji, T. 2016. Pengertian *VCC* dan *Ground* dalam Elektronika dan *Arduino*. Http://www.rustamaji.net/id/arduino/pengertian-vcc-dan-ground-dalam-elektronika-dan-arduino. Diakses 15 Oktober 2021
Mirul. 2017. Sensor gas semikonduktor dan cara kerjanya. http://k-sience.blogspot.com/2017/07/sensor-gas-semikonduktor-dan-cara.html. Diakses 15 Oktober 2021.

LAMPIRAN

```
#code program phyton
#import library
import numpy as np
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
import time
import sklearn

#data
data = pd.read_csv('D:\AMF\PROPOSAL\FIX PRPOSAL\daftar
    persamaan\jurnal pro\ISITIA 23\dataset gerd\data3sensor.csv')
data.head(5)

#klasifikasi data
class_A_data = data[data['Class'] == 'mild']  # Memilih data dengan
    kelas Mild
class_B_data = data[data['Class'] == 'moderate']  # Memilih data
    dengan kelas Moderate
class_C_data = data[data['Class'] == 'severe']  # Memilih data dengan
    kelas Severe
    _data = data[data['Class'] == 'healthy']  # Memilih data dengan
    as Healthy
```

```
# mengambil % dari semua kelas untuk training, serta sisa untuk
    testing
train_size = 0.8
test_size = 1 - train_size

class_A_train, class_A_test = train_test_split(class_A_data,
    test_size=test_size, random_state=42)
class_B_train, class_B_test = train_test_split(class_B_data,
    test_size=test_size, random_state=42)
class_C_train, class_C_test = train_test_split(class_C_data,
    test_size=test_size, random_state=42)
class_D_train, class_D_test = train_test_split(class_D_data,
    test_size=test_size, random_state=42)

# Menggabungkan data training dan testing untuk setiap kelas
    menjadi subset training dan testing secara keseluruhan
training_data = pd.concat([class_A_train, class_B_train,
    class_C_train, class_D_train], ignore_index=True)
testing_data = pd.concat([class_A_test, class_B_test, class_C_test,
    class_D_test], ignore_index=True)

# Memisahkan fitur (X) dan target (y) dari subset training dan
    testing
X_train = training_data[["MQ-135","MQ-136","MQ-7"]]
        = training_data['Class']


        = testing_data[["MQ-135","MQ-136","MQ-7"]]
```

```
y_test = testing_data['Class']

%%time
from sklearn.neighbors import KNeighborsClassifier
# Inisialisasi model KNN dengan jumlah tetangga (n_neighbors) yang
    diinginkan
knn = KNeighborsClassifier(n_neighbors=5)  # Misalnya,
    n_neighbors=5

# Melatih model pada data training
knn.fit(X_train, y_train)

# Melakukan prediksi pada data testing
predictions = knn.predict(X_test)

# Mengukur kinerja model (misalnya, menggunakan metrik
    akurasi untuk klasifikasi)
from sklearn.metrics import accuracy_score

accuracy = accuracy_score(y_test, predictions)
print(f"Akurasi model: {accuracy}")

# Menampilkan classification report yang mencakup precision,
    recall, dan f1-score
report = classification_report(y_test, predictions)
lassification Report:")
ort)
```

```
error_rate = []

# Will take some time
for i in range(1,10):

 knn = KNeighborsClassifier(n_neighbors=i)
 knn.fit(X_train,y_train)
 pred_i = knn.predict(X_test)
 error_rate.append(np.mean(pred_i != y_test))
plt.figure(figsize=(10,6))
plt.plot(range(1,10),error_rate,color='blue', linestyle='dashed',
        marker='o',
 markerfacecolor='red', markersize=10)
plt.title('Error Rate vs. K Value')
plt.xlabel('K')
plt.ylabel('Error Rate')

accuracy_scores = []

for k in range (1,10):
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train, y_train)
    y_pred = knn.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    accuracy_scores.append(accuracy)
```

ıg the results

e(figsize=(10, 6))

```
plt.plot(accuracy_scores, marker='o', linestyle='-', color='b')
plt.title('KNN Model Performance with Different Values of K')
plt.xlabel('Number of Neighbors (K)')
plt.ylabel('Accuracy')
plt.grid(True)
plt.show()
```

```
#code program arduino IDE
#include <LiquidCrystal_I2C.h>

//Include the library
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>
#include <WiFi.h>
#include <FirebaseESP32.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>


#include <MQUnifiedsensor.h>

//Definitions
#define placa "Arduino UNO"
#define Voltage_Resolution 5
#define pin 25 //Analog input 0 of your arduino
#define type "MQ-136" //MQ136
#define ADC_Bit_Resolution 10 // For arduino UNO/MEGA/NANO
#define RatioMQ136CleanAir 3.6//RS / R0 = 3.6 ppm
//#define calibration_button 13 //Pin to calibrate your sensor

//Declare Sensor
MQUnifiedsensor MQ136(placa, Voltage_Resolution, ADC_Bit_Resolution,
pin, type);
#define DHTPIN 4
#define VOLTAGE 5
#define DHTTYPE    DHT11     // DHT 11
#define WIFI_SSID "LP"
#define WIFI_PASSWORD "12345678"
#define RL 10 //nilai RL =10 pada sensor
#define m -0.417 //hasil perhitungan gradien
#define b 0.425 //hasil perhitungan perpotongan
#define Ro 19 //hasil pengukuran RO
#define MQ_sensor 26 //definisi variabel
#define placa "Arduino UNO"
#define Voltage_Resolution 5
#define pin 27 //Analog input 0 of your arduino
#define type "MQ-7" //MQ7
ADC_Bit_Resolution 10 // For arduino UNO/MEGA/NANO
RatioMQ7CleanAir 27.5 //RS / R0 = 27.5 ppm
PWMPin 5 // Pin connected to mosfet
```

```
DHT_Unified dht(DHTPIN, DHTTYPE);
uint32_t delayMS;
const int tombol = 32;
LiquidCrystal_I2C lcd(0x27, 16, 2); //adrressing lcd 0x27,16X2 lcd

const int numReadings = 5;//nilai penambilan sample pembacaan
sebesar 5 kali
float readings[numReadings];
int readIndex = 0;
float total = 0;
float average = 0;
const int relay = 33;
float hasil = 0;
float hasil136 = 0;
float hasil7 = 0;
MQUnifiedsensor MQ7(placa, Voltage_Resolution, ADC_Bit_Resolution,
pin, type);
unsigned long oldTime = 0;

void setup() {
  pinMode(relay, OUTPUT);
  //Init the serial port communication - to debug the library
  Serial.begin(115200); //Init serial port
  lcd.init();
  lcd.backlight();
  pinMode(tombol, INPUT);
  //  wifiset();
  dhtset();
  mq135();
  mq7();
}

void loop() {
  lcd.backlight();
  hasil7 = 0;
  int scan = 0;
  scan = digitalRead(tombol);
  Serial.println(scan);
  if (scan == 1) {
    scanning();

  } else {
      mq7();
      ata();


      VRL;
```

```
  float RS;
  float ratio;

  VRL = analogRead(MQ_sensor) * (5 / 1023.0); //konversi nilai
analog ke tegangan
  RS = (5.0 / VRL - 1) * 10 ; //RS = ((VC/VRL)-1)*RL (rumus untuk
mencari nilai RS)
  ratio = RS / Ro; // Rumus untuk mencari ratio (RO adalah 3.6
seperti yang didapatkan di datasheet sensor)
  float ppm = pow(10, ((log10(ratio) - b) / m)); //rumus mencari ppm
/ konsentrasi Gas

  total = total - readings[readIndex];

  readings[readIndex] = ppm;

  total = total + readings[readIndex];

  readIndex = readIndex + 1;


  if (readIndex >= numReadings) {

    readIndex = 0;
  }

  delay (1000);
  lcd.clear();
  average = total / numReadings;

}
```