LAMPIRAN

1. Program Arduino

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <DHT.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);
const int mq135Pin = A0;
const int dhtPin = 4;
const int buzzerPin = 8;

const int relayFan1Pin = 9;
const int relayFan2Pin = 10;

const int ledRedPin = 11;
const int ledGreenPin = 12;
const int ledBluePin = 13;

const float RL = 10.0;
const float RO_CLEAN_AIR_FACTOR = 9.83;

float RO = 10.0;

DHT dht(dhtPin, DHT22); // Change DHT11 to DHT22

float MQ135_Kalibrasi();
float MQ135_DapatkanResistansi(int raw_adc);
float MQ135_DapatkanPPM(float ratio);
void scrollTeks(String teks);
float kompensasiSuhu(float suhu);
float kompensasiKelembapan(float kelembapan);

void setup() {
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  lcd.print("MQ135 & DHT22"); // Update display to DHT22
  delay(2000);
  lcd.clear();

  dht.begin();
  RO = MQ135_Kalibrasi();
```

```
  Serial.print("RO: ");
  Serial.println(RO);

  pinMode(buzzerPin, OUTPUT);
  pinMode(relayFan1Pin, OUTPUT);
  pinMode(relayFan2Pin, OUTPUT);
  pinMode(ledRedPin, OUTPUT);
  pinMode(ledGreenPin, OUTPUT);
  pinMode(ledBluePin, OUTPUT);

  // Pastikan relay dan buzzer mati pada awalnya
  digitalWrite(buzzerPin, LOW);
  digitalWrite(relayFan1Pin, HIGH);
  digitalWrite(relayFan2Pin, HIGH);

  digitalWrite(ledRedPin, HIGH);
  digitalWrite(ledGreenPin, HIGH);
  digitalWrite(ledBluePin, HIGH);
}

void loop() {
  int sensorValue = analogRead(mq135Pin);
  float rs = MQ135_DapatkanResistansi(sensorValue);
  float ratio = rs / RO;
  float nh3_ppm = MQ135_DapatkanPPM(ratio);

  float kelembapan = dht.readHumidity();
  float suhu = dht.readTemperature();

  // Kompensasi suhu dan kelembapan
  float nh3_ppm_terkompensasi = nh3_ppm *
kompensasiSuhu(suhu) * kompensasiKelembapan(kelembapan);

  String teksTampilan = "Suhu: " + String(suhu) + "C
Kelembapan: " + String(kelembapan) + "%";
  scrollTeks(teksTampilan);

  lcd.setCursor(0, 0);
  lcd.print("NH3 PPM:");
  lcd.print(nh3_ppm_terkompensasi);

  Serial.print("NH3 PPM: ");
  Serial.println(nh3_ppm_terkompensasi);
  Serial.print("S: ");
  Serial.println(suhu);
```

```
  Serial.print("K: ");
  Serial.println(kelembapan);

  // Logika baru untuk NH3 > 100 ppm
  if (nh3_ppm_terkompensasi > 100.0) {
    Serial.println("NH3 > 100.0 ppm, BERBAHAYA!");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("BERBAHAYA!");
    lcd.setCursor(0, 1);
    lcd.print("NH3 > 100 ppm");

    digitalWrite(ledRedPin, HIGH);
    digitalWrite(ledGreenPin, LOW);
    digitalWrite(ledBluePin, LOW);

    digitalWrite(relayFan1Pin, LOW);  // Aktifkan relay
(LOW karena relay aktif rendah)
    digitalWrite(relayFan2Pin, LOW);  // Aktifkan relay
(LOW karena relay aktif rendah)

    tone(buzzerPin, 2500);  // Bip terus menerus pada
2500 Hz
  } else if (nh3_ppm_terkompensasi > 5.0) {
    Serial.println("NH3 > 5.0 ppm, mengaktifkan LED
merah, kipas, dan buzzer...");
    digitalWrite(ledRedPin, HIGH);
    digitalWrite(ledGreenPin, LOW);
    digitalWrite(ledBluePin, LOW);
    // Aktifkan relay untuk kipas 1 dan kipas 2
    digitalWrite(relayFan1Pin, LOW);  // Aktifkan relay
(LOW karena relay aktif rendah)
    digitalWrite(relayFan2Pin, LOW);  // Aktifkan relay
(LOW karena relay aktif rendah)

    // Bunyi bip
    tone(buzzerPin, 2500);  // Frekuensi 2500 Hz
    delay(1000);  // Bunyi selama 1000 ms
    noTone(buzzerPin);  // Matikan bunyi
    delay(1000);  // Hening selama 1000 ms
  } else if (suhu > 33.0) {
    Serial.println("Suhu > 33.0C, mengaktifkan LED merah,
kipas, dan buzzer...");

    // LED merah
```

```
    digitalWrite(ledRedPin, HIGH);
    digitalWrite(ledGreenPin, LOW);
    digitalWrite(ledBluePin, LOW);

    // Bunyi bip
    for (int i = 0; i < 5; i++) {
      tone(buzzerPin, 2500);  // Frekuensi 2500 Hz
      analogWrite(buzzerPin, 255);
      delay(200);  // Bunyi selama 200 ms
      noTone(buzzerPin);  // Matikan bunyi
      delay(200);  // Hening selama 200 ms
      tone(buzzerPin, 2500);
      analogWrite(buzzerPin, 255);
      delay(200);
      noTone(buzzerPin);
      delay(200);
      tone(buzzerPin, 2500);
      analogWrite(buzzerPin, 255);
      delay(200);
      noTone(buzzerPin);
      delay(200);
    }

    // Aktifkan relay untuk kipas 1 dan kipas 2
    digitalWrite(relayFan1Pin, LOW);
    digitalWrite(relayFan2Pin, LOW);

  } else if (suhu > 30.0) {
    Serial.println("Suhu > 30.0C, mengaktifkan LED kuning
dan kipas...");

    // LED kuning (Merah + Hijau)
    digitalWrite(ledRedPin, HIGH);
    digitalWrite(ledGreenPin, HIGH);
    digitalWrite(ledBluePin, LOW);

    // Nonaktifkan buzzer
    noTone(buzzerPin);

    // Aktifkan relay untuk kipas 1 dan kipas 2
    digitalWrite(relayFan1Pin, LOW);
    digitalWrite(relayFan2Pin, LOW);

  } else if (suhu > 20.0) {
```

```
    Serial.println("Suhu > 20.0C, mengaktifkan LED
hijau...");

    // LED hijau
    digitalWrite(ledRedPin, LOW);
    digitalWrite(ledGreenPin, HIGH);
    digitalWrite(ledBluePin, LOW);

    // Nonaktifkan buzzer
    noTone(buzzerPin);

    // Matikan relay untuk kipas 1 dan kipas 2
    digitalWrite(relayFan1Pin, HIGH);
    digitalWrite(relayFan2Pin, HIGH);
  } else {

    Serial.println("Suhu > 28.0C, mengaktifkan LED
kuning...");

    // Matikan semua LED
    digitalWrite(ledRedPin, LOW);
    digitalWrite(ledGreenPin, LOW);
    digitalWrite(ledBluePin, HIGH);

    // Nonaktifkan buzzer
    noTone(buzzerPin);

    // Matikan relay untuk kipas 1 dan kipas 2
    digitalWrite(relayFan1Pin, HIGH);
    digitalWrite(relayFan2Pin, HIGH);
  }

  delay(2000);
}

float MQ135_Kalibrasi() {
  float val = 0;
  for (int i = 0; i < 50; i++) {
    val +=
MQ135_DapatkanResistansi(analogRead(mq135Pin));
    delay(100);
  }
  val = val / 50;
  val = val / RO_CLEAN_AIR_FACTOR;
  return val;
```

```
}

float MQ135_DapatkanResistansi(int raw_adc) {
  return ((1023.0 / raw_adc) - 1.0) * RL;
}

float MQ135_DapatkanPPM(float ratio) {
  const float a = 102.2;
  const float b = -2.221;

  return a * pow(ratio, b);
}

void scrollTeks(String teks) {
  int len = teks.length();
  for (int i = 0; i < len + 16; i++) {
    lcd.setCursor(0, 1);
    if (i < len) {
      lcd.print(teks.substring(i, i + 16));
    } else {
      lcd.print(teks.substring(0, 16 - (i - len)));
    }
    delay(300);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("NH3 PPM: ");
    lcd.print(MQ135_DapatkanPPM(MQ135_DapatkanResistansi(
analogRead(mq135Pin)) / RO));
  }
}

float kompensasiSuhu(float suhu) {
  // Kompensasi berdasarkan suhu (contoh sederhana)
  if (suhu < 20) return 1.1;
  else if (suhu > 30) return 0.9;
  else return 1.0;
}

float kompensasiKelembapan(float kelembapan) {
  // Kompensasi berdasarkan kelembapan (contoh sederhana)
  if (kelembapan < 30) return 1.1;
  else if (kelembapan > 70) return 0.9;
  else return 1.0;
}
```