## LAMPIRAN

A. Source Code Alat

```
#include <ESP8266WiFi.h>
#include <LiquidCrystal_I2C.h>
#include "DHT.h"
#include <Wire.h>
#include "ThingSpeak.h"

const char* ssid = "realme C61";
const char* password = "lili1709";
unsigned long channelID = 2926723;
const char* writeAPIKey = "Q9CSKORR6P3KGUF2";

#define DHTPIN D6
#define DHTTYPE DHT22
#define MQ2_PIN D7
#define DUST_SENSOR_PIN D5
#define MQ135_PIN A0
#define BUZZER_PIN D8

DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 16, 2);
WiFiClient client;

const int MQ135_AMAN = 200, MQ135_TERCEMAR = 400;
const int DEBU_AMAN = 100, DEBU_TERCEMAR = 300;
const float SUHU_MIN = 20.0, SUHU_MAX = 30.0;
const float KELEMBABAN_MIN = 40.0, KELEMBABAN_MAX = 70.0;

enum Status { AMAN = 0, WASPADA = 1, BAHAYA = 2 };

void setup() {
  Serial.begin(115200);
  Wire.begin(D2, D1);
  lcd.begin(16, 2);
  lcd.backlight();
  dht.begin();
  pinMode(MQ2_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);

  lcd.print("Connecting WiFi");
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
```

```
  ThingSpeak.begin(client);
  lcd.clear();
  lcd.print("WiFi Connected!");
  delay(1500);
}

Status evaluasi(int nilai, int aman, int tercemar) {
  if (nilai < aman) return AMAN;
  else if (nilai < tercemar) return WASPADA;
  else return BAHAYA;
}

Status evaluasiSuhu(float nilai) {
  if (nilai >= SUHU_MIN && nilai <= SUHU_MAX) return AMAN;
  else if ((nilai >= SUHU_MIN-5 && nilai < SUHU_MIN) || (nilai > SUHU_MAX
&& nilai <= SUHU_MAX+5)) return WASPADA;
  else return BAHAYA;
}

Status evaluasiKelembaban(float nilai) {
  if (nilai >= KELEMBABAN_MIN && nilai <= KELEMBABAN_MAX) return
AMAN;
  else if ((nilai >= KELEMBABAN_MIN-10 && nilai < KELEMBABAN_MIN)
|| (nilai > KELEMBABAN_MAX && nilai <= KELEMBABAN_MAX+15))
return WASPADA;
  else return BAHAYA;
}

Status getStatusKeseluruhan(Status mq135, Status debu, Status suhu, Status
kelembaban, bool mq2) {
  if (mq2) return BAHAYA;
  Status max = AMAN;
  if (mq135 > max) max = mq135;
  if (debu > max) max = debu;
  if (suhu > max) max = suhu;
  if (kelembaban > max) max = kelembaban;
  return max;
}

String statusStr(Status s) {
  switch (s) {
    case AMAN: return "AMAN";
    case WASPADA: return "WASPADA";
    case BAHAYA: return "BAHAYA";
    default: return "ERROR";
  }
}
```

```
void alarm(Status s) {
  switch (s) {
    case WASPADA:
      digitalWrite(BUZZER_PIN, HIGH);
      delay(200);
      digitalWrite(BUZZER_PIN, LOW);
      break;
    case BAHAYA:
      for (int i = 0; i < 3; i++) {
        digitalWrite(BUZZER_PIN, HIGH);
        delay(500);
        digitalWrite(BUZZER_PIN, LOW);
        delay(200);
      }
      break;
  }
}

void loop() {
  float suhu = dht.readTemperature();
  float kelembaban = dht.readHumidity();
  int mq135 = analogRead(MQ135_PIN);
  int debu = analogRead(DUST_SENSOR_PIN);
  bool mq2 = (digitalRead(MQ2_PIN) == HIGH);

  if (isnan(suhu) || isnan(kelembaban)) {
    lcd.clear();
    lcd.print("Error DHT22!");
    delay(2000);
    return;
  }

  Status sMQ135 = evaluasi(mq135, MQ135_AMAN, MQ135_TERCEMAR);
  Status sDebu = evaluasi(debu, DEBU_AMAN, DEBU_TERCEMAR);
  Status sSuhu = evaluasiSuhu(suhu);
  Status sKelembaban = evaluasiKelembaban(kelembaban);
    Status sKeseluruhan = getStatusKeseluruhan(sMQ135, sDebu, sSuhu,
sKelembaban, mq2);

  alarm(sKeseluruhan);

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("STATUS:");
  lcd.setCursor(0, 1);
  lcd.print(statusStr(sKeseluruhan));
  if (sKeseluruhan == BAHAYA) lcd.print(" !!!");
  delay(3000);
```

```
  lcd.clear();
  lcd.print("T:" + String(suhu, 1) + "C " + statusStr(sSuhu).substring(0,3));
  lcd.setCursor(0, 1);
      lcd.print("H:" + String(kelembaban, 1) + "% " +
statusStr(sKelembaban).substring(0,3));
  delay(3000);

  lcd.clear();
  lcd.print("Gas:");
  lcd.setCursor(0, 1);
  lcd.print(mq2 ? "TERDETEKSI!" : "TIDAK ADA");
  delay(3000);

  lcd.clear();
  lcd.print("Air:" + String(mq135) + " " + statusStr(sMQ135).substring(0,3));
  lcd.setCursor(0, 1);
  lcd.print("Debu:" + String(debu) + " " + statusStr(sDebu).substring(0,3));
  delay(3000);

  Serial.println("T:" + String(suhu) + "C H:" + String(kelembaban) + "% MQ135:"
+ String(mq135) + " Debu:" + String(debu) + " MQ2:" + String(mq2) + " Status:"
+ statusStr(sKeseluruhan));

  ThingSpeak.setField(1, suhu);
  ThingSpeak.setField(2, kelembaban);
  ThingSpeak.setField(3, mq135);
  ThingSpeak.setField(4, debu);
  ThingSpeak.setField(5, mq2);
  ThingSpeak.setField(6, (int)sKeseluruhan);

  int status = ThingSpeak.writeFields(channelID, writeAPIKey);
  Serial.println(status == 200 ? "ThingSpeak OK" : "ThingSpeak Failed");
}
```