

```

#include <WiFi.h> // Library untuk koneksi WiFi
#include <WiFiClientSecure.h> // Library untuk koneksi HTTPS (Telegram)
#include <Firebase_ESP_Client.h> // Library Firebase ESP32
#include <math.h> // Library untuk perhitungan matematika (pow)

// ===== WIFI =====
#define WIFI_SSID "ISI_WIFI" // Ganti dengan nama WiFi-mu
#define WIFI_PASSWORD "ISI_PASSWORD" // Ganti dengan password WiFi-mu

// ===== FIREBASE =====
#define API_KEY "ISI_API_KEY" // Ganti dengan API Key Firebase
#define DATABASE_URL "ISI_DATABASE_URL" // Ganti dengan URL Realtime Database Firebase

FirebaseData fbdo; // Objek untuk komunikasi dengan Firebase
FirebaseAuth auth; // Auth object untuk Firebase
FirebaseConfig config; // Konfigurasi Firebase

// ===== TELEGRAM =====
String botToken = "ISI_BotToken"; // Ganti dengan token bot Telegram
String chatID = "ISI_ChatID"; // Ganti dengan chat ID Telegram
WiFiClientSecure client; // Client HTTPS untuk Telegram

// ===== SENSOR =====
int gasPin = 4; // Pin ADC ESP32 yang terhubung ke MQ135

// ===== KALIBRASI SENSOR =====
float R0 = 120.0; // Nilai sensor MQ135 pada udara bersih (ubah sesuai kalibrasi)

// ===== THRESHOLD GAS BERBAHAYA =====
float batasPPM = 2000; // Ambang batas PPM gas yang dianggap bahaya

// ===== ANTI SPAM TELEGRAM =====
unsigned long lastSend = 0; // Waktu terakhir mengirim pesan
unsigned long interval = 10000; // Interval minimal antar pesan (10 detik)

// ===== FUNCTION KIRIM TELEGRAM =====
void sendTelegram(String text){
    client.setInsecure(); // Disable certificate check (ESP32 terbatas)

    // Cek koneksi ke server Telegram
    if (!client.connect("api.telegram.org", 443)){
        Serial.println("Telegram gagal konek");
        return;
    }

    // Ganti spasi dengan %20 untuk URL
    text.replace(" ", "%20");

    // Buat URL untuk kirim pesan
    String url = "/bot" + botToken +
        "/sendMessage?chat_id=" + chatID +
        "&text=" + text;

    // Kirim request GET ke Telegram
    client.print(String("GET ") + url + " HTTP/1.1\r\n" +
        "Host: api.telegram.org\r\n" +
        "Connection: close\r\n\r\n");
}

// ===== SETUP =====
void setup(){
    Serial.begin(115200); // Mulai Serial Monitor untuk debugging

    // ===== CONNECT KE WIFI =====
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
    while (WiFi.status() != WL_CONNECTED){
        delay(500);
        Serial.print("."); // Menampilkan loading koneksi
    }
    Serial.println("\nWiFi OK");

    // ===== SETUP FIREBASE =====
    config.api_key = API_KEY;

```

```

config.database_url = DATABASE_URL;

// Sign up anonymous user & inisialisasi Firebase
Firebase.signUp(&config, &auth, "", "");
Firebase.begin(&config, &auth);
Firebase.reconnectWiFi(true);

Serial.println("Firebase Ready");

// Kirim notifikasi Telegram bahwa ESP32 sudah online
sendTelegram("ESP32 online");
}

// ===== LOOP UTAMA =====
void loop(){

// ===== BACA SENSOR ADC DENGAN RATA-RATA =====
long total = 0;
for(int i=0;i<10;i++){
    total += analogRead(gasPin); // Baca sensor MQ135
    delay(50); // Delay antar pembacaan
}
float adc = total / 10.0; // Hitung rata-rata ADC

// ===== HITUNG PPM GAS =====
float ratio = adc / R0; // Rasio terhadap kalibrasi
float ppm = 116.6020682 * pow(ratio, -2.769034857); // Rumus MQ135

// Tampilkan hasil di Serial Monitor
Serial.print("ADC: ");
Serial.print(adc);
Serial.print(" | PPM: ");
Serial.println(ppm);

// ===== KIRIM DATA KE FIREBASE =====
Firebase.RTDB.setFloat(&fbdo, "/ppm", ppm);
Firebase.RTDB.setFloat(&fbdo, "/gas", ppm);

// ===== ALERT TELEGRAM JIKA GAS BAHAYA =====
if(ppm > batasPPM && millis() - lastSend > interval){
    sendTelegram("GAS BAHAYA! PPM: " + String(ppm));
    lastSend = millis(); // Update waktu terakhir kirim
}

delay(2000); // Delay loop utama
}

```