

**PRATIKUM**  
**SISTEM IOT UNTUK MONITORING SUHU, KELEMBAPAN,**  
**DAN DETEKSI API MENGGUNAKAN ESP8266 DAN**  
**PLATFORM TELEGRAM**

Mata Kuliah Sistem Berbasis Internet Of Things

Dosen Pengampu : Solichudin, S.Pd., M.T.



**Disusun Oleh :**

Wildan Nur Yusufi	(2208096045)
Muhammad Lukmanul Khakim	(2208096068)
Irvan Nurmutakim	(2208096070)

**PROGRAM STUDI TEKNOLOGI INFORMASI**  
**FAKULTAS SAINS DAN TEKNOLOGI**  
**UNIVERSITAS ISLAM NEGERI WALISONGO**  
**SEMARANG**

**2025**

# BAB I

## TINJAUAN PUSTAKA

### 1. Suhu dan Kelembaban

Suhu adalah ukuran tingkat panas atau dinginnya suatu benda atau lingkungan. Suhu menunjukkan energi panas yang dimiliki oleh suatu objek atau udara. Semakin tinggi suhu, semakin panas suatu benda atau ruangan tersebut. Memiliki satuan Derajat Celsius ( $^{\circ}\text{C}$ ), Fahrenheit ( $^{\circ}\text{F}$ ), atau Kelvin (K) dan alat ukur suhu berupa Termometer atau sensor suhu seperti DHT11, DHT22, atau DS18B20.

Kelembaban adalah jumlah uap air yang ada di udara. Kelembaban udara menunjukkan seberapa basah atau kering udara di sekitarnya. Biasanya dinyatakan dalam bentuk kelembaban relatif (Relative Humidity atau %RH), yaitu perbandingan antara jumlah uap air yang ada saat ini dengan jumlah maksimum yang bisa ditampung udara pada suhu tertentu. Memiliki satuan Persentase (%RH – Relative Humidity) dan alat ukur berupa Higrometer atau sensor kelembaban seperti DHT11 dan DHT22.

### 2. ESP8266

ESP8266 merupakan perangkat elektronik yang dapat menggunakan mikrokontroler dan antarmuka jaringan, juga memiliki banyak pin I/O, dapat dikembangkan untuk mengontrol atau memantau aplikasi IoT. NodeMCU mirip dengan pengontrol Arduino ESP8266. Pemrograman board ini memerlukan banyak teknik manual dan penambahan USB ke mode serial, yang sulit dan memerlukan keterampilan pemrograman. Namun, NodeMCU menyertakan multi-kontroler ESP8266 onboard yang memungkinkan akses jaringan dan komunikasi USB on-chip.



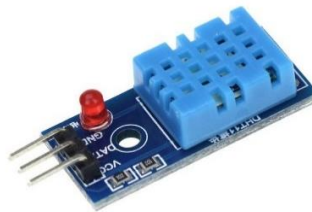
*Gambar 1. ESP8266*

Hal ini menyederhanakan aplikasi karena hanya memerlukan kabel data USB. NodeMCU ESP8266 berbeda dari semua board sebelumnya karena NodeMCU ESP8266 menggunakan driver chip FTDI USB-to-serial. Sebaliknya, ia menggunakan fungsionalitas

CHIP WIFI yang dirancang sebagai adaptor USB ke server. Papan NodeMCU ESP8266 memiliki fitur baru berikut : 1. Pinout 2. Pemulihan sirkuit 3. ATmega16U2 menggantikan 8U yang digunakan sebagai konverter USB ke server.

### 3. DHT11

Sensor digital yang digunakan untuk mengukur suhu dan kelembaban udara. Sensor ini banyak digunakan dalam berbagai proyek elektronika dan Internet of Things (IoT) karena harganya terjangkau, mudah digunakan, dan kompatibel dengan mikrokontroler seperti Arduino, ESP8266, dan ESP32. DHT11 mampu mengukur suhu dalam rentang 0°C hingga 50°C dengan akurasi  $\pm 2^{\circ}\text{C}$ , serta kelembaban relatif antara 20% hingga 90% dengan akurasi  $\pm 5\%$  RH.



*Gambar 2. DHT11*

Sensor ini menghasilkan data dalam bentuk sinyal digital, sehingga memudahkan proses pembacaan oleh mikrokontroler. DHT11 bekerja dengan tegangan antara 3.3V hingga 5.5V dan memiliki kecepatan sampling sekitar satu kali per detik (1Hz). Meskipun memiliki kelebihan seperti konsumsi daya rendah dan kemudahan penggunaan, DHT11 memiliki keterbatasan dalam hal akurasi dan rentang pengukuran jika dibandingkan dengan versi yang lebih canggih seperti DHT22. Sensor ini sangat cocok digunakan untuk pemantauan lingkungan sederhana, seperti sistem monitoring suhu dan kelembaban ruangan, rumah pintar, atau green house yang terhubung dengan notifikasi melalui aplikasi seperti Telegram.

### 4. Sensor MQ2

Sensor MQ2 adalah sensor gas analog yang digunakan untuk mendeteksi keberadaan berbagai jenis gas mudah terbakar dan berbahaya di udara, seperti LPG, butana, propana, metana, hidrogen, asap, dan alkohol. Sensor ini bekerja dengan prinsip perubahan resistansi bahan semikonduktor ( $\text{SnO}_2$ ) yang sensitif terhadap gas. Ketika gas yang terdeteksi masuk ke dalam sensor, resistansi internalnya akan berubah, menghasilkan tegangan keluaran yang bisa dibaca oleh mikrokontroler seperti Arduino.



*Gambar 3. Sensor MQ2*

MQ2 memiliki dua jenis output: **analog** dan **digital**. Output analog memberikan nilai tegangan yang proporsional dengan konsentrasi gas, sedangkan output digital (melalui modul dengan komparator) akan mengeluarkan sinyal HIGH atau LOW tergantung apakah nilai ambang batas gas telah terlampaui. Sensor ini sering digunakan dalam sistem deteksi kebocoran gas, sistem keamanan rumah, atau proyek IoT untuk pemantauan kualitas udara karena harganya yang terjangkau dan kemudahannya untuk diintegrasikan dengan sistem elektronik.

#### 5. Power Supply

Power supply atau catu daya adalah perangkat yang berfungsi untuk menyediakan dan mengatur tegangan serta arus listrik yang dibutuhkan oleh suatu sistem elektronik atau perangkat listrik agar dapat beroperasi dengan baik.

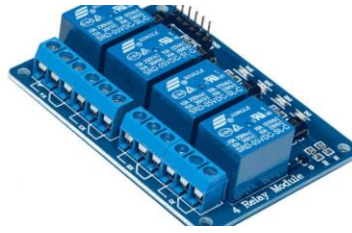


*Gambar 5. Power Supply*

Power supply mengubah sumber energi listrik dari satu bentuk ke bentuk lain yang sesuai, misalnya dari arus listrik AC (seperti dari stopkontak rumah) menjadi arus DC (yang dibutuhkan oleh komponen elektronik seperti mikrokontroler, sensor, dan motor).

#### 6. Relay

Komponen elektromekanis yang berfungsi sebagai saklar elektronik yang dikendalikan oleh arus listrik bertegangan rendah untuk mengontrol arus yang lebih tinggi. Relay terdiri dari kumparan (coil), kontak saklar, dan mekanisme elektromagnetik.



*Gambar 6. Relay*

Ketika arus mengalir melalui kumparan, medan magnet yang terbentuk akan menarik saklar sehingga terjadi perubahan posisi kontak—dari terbuka (NC, Normally Closed) menjadi tertutup (NO, Normally Open), atau sebaliknya. Dengan mekanisme ini, relay memungkinkan mikrokontroler seperti ESP8266, yang hanya menghasilkan sinyal kecil, untuk mengontrol perangkat listrik bertegangan tinggi seperti lampu, kipas, pompa, atau bahkan alat rumah tangga lainnya. Relay banyak digunakan dalam sistem otomatisasi rumah (smart home), sistem IoT, dan pengendalian beban listrik karena keamanannya dalam memisahkan sirkuit kontrol bertegangan rendah dari sirkuit daya bertegangan tinggi.

## 7. LCD12C

LCD I2C adalah modul tampilan Liquid Crystal Display (LCD) yang menggunakan antarmuka I2C (Inter-Integrated Circuit) untuk berkomunikasi dengan mikrokontroler seperti Arduino atau ESP8266. Umumnya, LCD I2C yang digunakan berukuran 16x2, yang berarti dapat menampilkan 2 baris dengan masing-masing 16 karakter.



*Gambar 7. LCD12C*

Keunggulan utama dari penggunaan I2C adalah penghematan jumlah pin yang digunakan; jika pada LCD biasa diperlukan setidaknya 6–8 pin digital, maka dengan I2C cukup menggunakan dua pin saja (SDA dan SCL), sehingga sisa pin dapat dimanfaatkan untuk komponen lain. Modul I2C biasanya merupakan adaptor kecil yang dipasang di bagian belakang LCD dan dilengkapi dengan chip PCF8574 yang mengatur komunikasi I2C. Selain mempermudah koneksi, modul ini juga sering dilengkapi dengan **potensiometer** untuk mengatur kecerahan layar. LCD I2C sangat berguna dalam berbagai

proyek mikrokontroler, terutama yang memerlukan tampilan informasi seperti suhu, kelembaban, status sensor, dan lainnya, karena praktis, hemat pin, dan mudah diprogram menggunakan pustaka (library) yang tersedia luas.

#### 8. Motor DC

Motor DC (Direct Current) adalah jenis motor listrik yang mengubah energi listrik arus searah (DC) menjadi energi gerak (mekanis) berupa putaran. Prinsip kerja motor DC didasarkan pada gaya Lorentz, yaitu ketika arus listrik mengalir melalui kumparan dalam medan magnet, maka akan timbul gaya yang menyebabkan rotor berputar.



*Gambar 8. Motor DC*

Motor DC terdiri dari beberapa komponen utama, seperti stator (bagian diam dengan magnet), rotor atau armature (bagian yang berputar), komutator, dan sikat (brush). Keunggulan motor DC adalah kemampuannya dalam mengatur kecepatan dan arah putaran dengan mudah melalui pengaturan tegangan atau polaritas arus. Motor ini banyak digunakan dalam berbagai perangkat elektronik dan sistem otomatisasi, seperti mainan elektrik, kipas, pompa, robotika, hingga kendaraan listrik.

#### 9. Telegram

Telegram adalah layanan pesan instan berbasis cloud yang gratis dan nirlaba. Telegram ini dapat digunakan oleh pengguna dengan sistem operasi berbeda seperti iOS, Windows Phone, Android, Ubuntu Touch, macOS, Linux, Windows.

##### Fitur Aplikasi Telegram

- a. Secret Chat Fitur ini untuk memberikan privasi berkomunikasi antar pengguna, dengan memanfaatkan teknologi client-to-client dengan MTproto sebagai protokol keamanannya, sehingga tidak menyimpan di server cloud telegram, berbeda dengan chat biasa telegram yang menyimpan semuanya di cloud.
- b. Grup Telegram Pada grup telegram ini dapat menampung sampai dengan 200 ribu anggota, sangat berbeda dengan grup whatsapp yang maksimal dapat

menampung 256 anggota. Oleh karena banyaknya daya tampung anggota  
JURNAL JUIT Vol 3 No. 2 Mei 2024 pISSN: 2828-6936, eISSN: 2828-6901,  
Page 17-27 20 pada grup telegram dapat memudahkan penggunaanya dalam  
mengelola komunitas/grup tersebut ataupun bisnis yang dikelola.

- c. Channel Telegram Pada September 2015 fitur ini telah diluncurkan untuk membangun sebuah komunitas tanpa batas untuk jumlah anggotanya. Memiliki fungsi yang mirip seperti grup akan tetapi hanya pembuat channel yang dapat mengirimkan pesan.
- d. BOT Telegram Telegram telah memunculkan platform untuk pada para pengembang pihak ketiga dalam membuat akun BOT, akun ini merupakan akun telegram yang dapat dijalankan pada sebuah program, seperti pengguna mengirimkan pesan ke BOT menggunakan perintah yang dapat dimengerti oleh BOT, sehingga dapat langsung dijawab pesan tersebut.

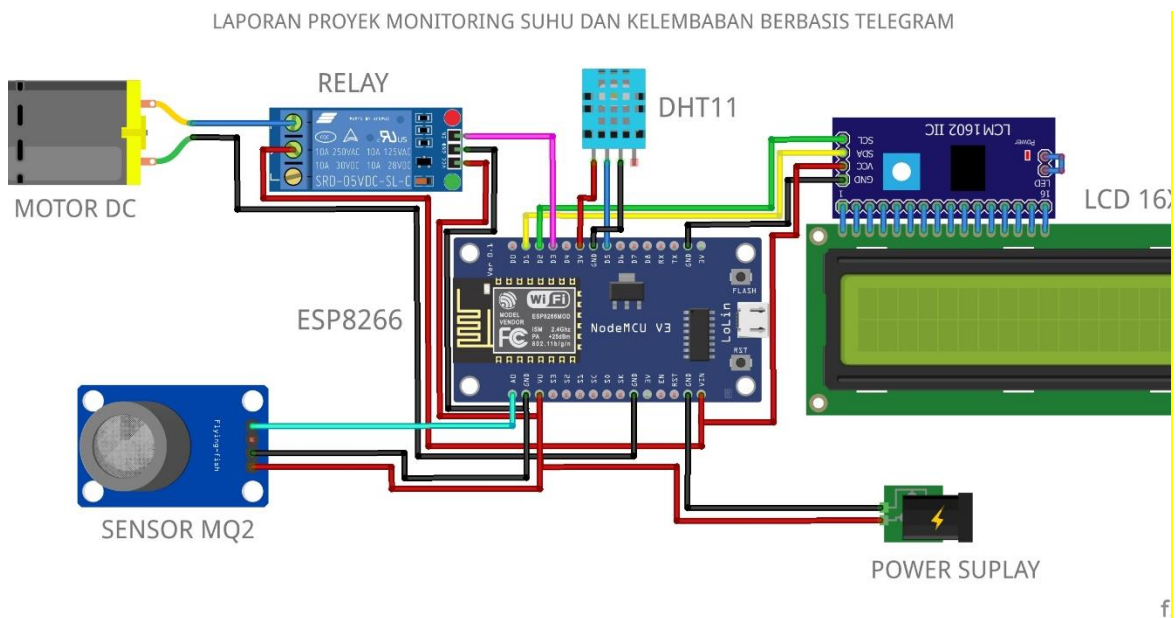
## BAB II

### HASIL DAN IMPLEMENTASI

#### A. TUJUAN PERCOBAAN

1. Mempelajari cara kerja sensor DHT11 dalam membaca data suhu dan kelembaban secara real-time.
2. Merancang dan membangun sistem monitoring berbasis IoT yang mampu mengirimkan data suhu dan kelembaban secara otomatis melalui platform Telegram.
3. Menumbuhkan pemahaman mahasiswa terhadap konsep dasar Internet of Things (IoT) serta penerapannya dalam kehidupan nyata, khususnya dalam pemantauan lingkungan.

#### B. DESAIN RANGKAIAN



Gambar 9. Desain Rangkaian

#### C. TABEL KABEL JUMPER

Tabel 1. Koneksi kabel Jumper

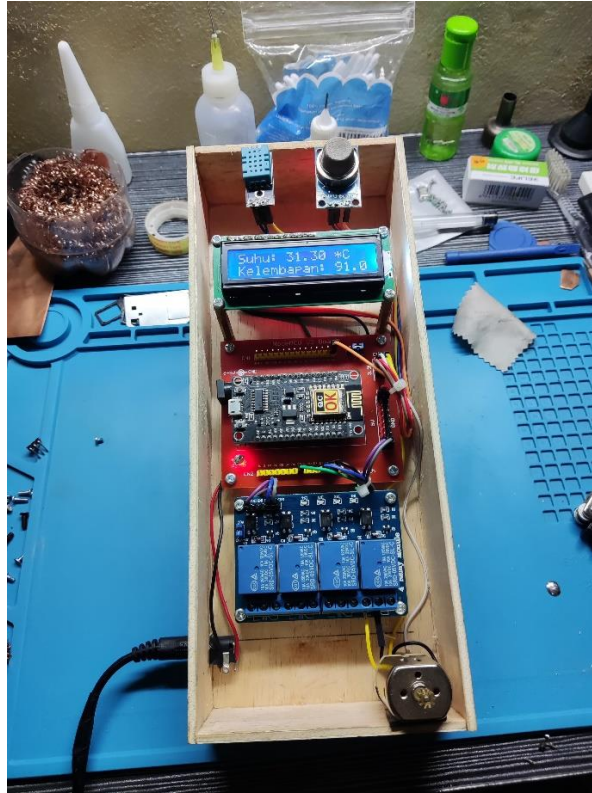
Komponen	Pin Pada Komponen	Koneksi Ke ESP8266	Keterangan
LCD I2C (16x2)	VCC	3V3	Sesuaikan dengan modul, umumnya aman 5V.



	GND	GND	Ground.
	SDA	D1	SDA untuk komunikasi I2C.
	SCL	D2	SCL untuk komunikasi I2C.
DHT11	VCC	3.3V	Sumber tegangan untuk sensor DHT11.
	GND	GND	Ground.
	OUT	D5	Pin data digital untuk membaca suhu & kelembapan.
Sensor MQ-2	VCC	3V3	Beberapa versi MQ-2 memerlukan 5V agar optimal.
	GND	GND	Ground.
	AO (Analog Output)	A0	Output analog gas (diukur oleh ESP8266).
Relay	VCC	3V3	Daya relay
	GND	GND	Ground relay
	IN	D1	Kontrol sinyal relay
Motor DC	(+)	NO (Relay)	Tegangan dikendalikan oleh relay
	(-)	GND Power Supply	Jalur ground motor
Power Supply	(+)	VIN / VCC modul Relay dan Motor	Tegangan eksternal 9V
	(-)	GND	Ground bersama (common ground)

## D. HASIL PERCOBAAN

### 1. Gambar Sistem



*Gambar 10. Sensor*

Sistem pada gambar merupakan sebuah alat monitoring suhu dan kelembaban berbasis mikrokontroler NodeMCU (ESP8266) yang dirakit secara manual dalam kotak kayu. Sistem ini menggunakan sensor DHT11 untuk membaca suhu dan kelembaban udara, lalu menampilkan data tersebut pada layar LCD 16x2 yang terpasang di bagian atas kotak. Terdapat juga joystick sebagai input tambahan yang kemungkinan digunakan untuk navigasi atau pengaturan manual. Di bagian tengah, NodeMCU berfungsi sebagai otak dari sistem, mengatur komunikasi antara sensor, LCD, dan modul lainnya. Di bagian bawah, terdapat modul relay 4-channel yang berfungsi untuk mengontrol perangkat eksternal secara otomatis, seperti kipas, pompa air, atau alat pendingin, berdasarkan data suhu dan kelembaban yang terbaca. Sistem ini juga terhubung dengan motor DC, yang dapat diaktifkan melalui salah satu relay. Dengan susunan komponen yang cukup rapi dan fungsi yang terintegrasi, alat ini cocok digunakan untuk keperluan otomasi lingkungan, seperti di rumah kaca, peternakan, atau ruang penyimpanan yang memerlukan kondisi udara tertentu.

## 2. Gambar Hasil Telegram



*Gambar 11. Hasil Telegram*

Gambar tersebut menampilkan hasil komunikasi antara sistem monitoring berbasis IoT dengan pengguna melalui aplikasi Telegram. Sistem ini dirancang untuk memantau suhu, kelembapan, dan keberadaan gas atau api secara real-time. Dari percakapan terlihat bahwa sistem secara otomatis mengirimkan peringatan ketika terdeteksi adanya gas berbahaya dan api, serta secara responsif mengaktifkan pompa air sebagai langkah pengamanan. Selain itu, pengguna dapat berinteraksi langsung dengan sistem menggunakan perintah seperti "Cek suhu", "Cek kelembapan", "Cek MQ", dan "Cek semua", yang akan dibalas oleh sistem dengan data terkini. Sebagai contoh, sistem merespons dengan mengirimkan data suhu 31,30°C, kelembapan 90,00%, dan nilai sensor gas MQ sebesar 234. Interaksi ini menunjukkan bahwa sistem tidak hanya bersifat otomatis, tetapi juga interaktif dan responsif terhadap perintah pengguna, menjadikannya sangat berguna untuk pemantauan lingkungan secara jarak jauh melalui Telegram.

## E. LINK VIDEO PERCOBAAN DAN GITHUB

Link Video Percobaan :

<https://youtu.be/-H3feliN37M>

Link Github Project Akhir

<https://github.com/irvan878/Kelompok4-IoT>

## F. KESIMPULAN

Program ini merupakan implementasi Sistem IoT berbasis ESP8266 untuk memantau suhu, kelembapan, dan keberadaan gas/api, serta memberikan notifikasi melalui Telegram secara real-time. Berikut kesimpulan dari fungsionalitas utama dalam kode tersebut::

### 1. Sensor DHT11

Digunakan untuk mengukur suhu dan kelembapan lingkungan secara berkala. Nilai yang terbaca ditampilkan di LCD 16x2 dan dapat dikirimkan melalui bot Telegram jika diminta.

### 2. Sensor Gas (MQ)

### 3. Membaca nilai gas atau asap di sekitar. Jika nilai melebihi ambang batas:

- $>600$  → Terjadi deteksi api → Relay diaktifkan (menyalakan pompa atau aktuator) dan pesan peringatan dikirim ke Telegram.
- $401-600$  → Gas ringan terdeteksi → Relay tetap mati, peringatan dikirim ke Telegram.
- $\leq 400$  → Normal, relay dimatikan.

### 4. Relay

Berfungsi mengontrol perangkat seperti pompa air sebagai respons otomatis jika deteksi kebakaran terjadi.

### 5. LCD I2C

Menampilkan status sistem seperti suhu, kelembapan, deteksi gas, nama kelompok, dan informasi jaringan.

### 6. Telegram Bot

Digunakan sebagai media komunikasi dua arah. Pengguna bisa mengirim perintah seperti:

- "Cek suhu"
- "Cek kelembapan"

- "Cek MQ"
- "Cek semua"

Bot akan membalas dengan informasi sensor secara real-time.

## 7. WiFi

Sistem menghubungkan ESP8266 ke jaringan Wi-Fi dengan SSID "Kelompok4" dan password "walisongo". Jika gagal, sistem akan menampilkan kredensial WiFi di LCD selama 8 detik dan mencoba kembali.

Fitur Tambahan dan Antarmuka:

- LCD akan menampilkan identitas kelompok, judul proyek, dan indikasi status sensor.
- Terdapat mekanisme fallback ketika Wi-Fi gagal terhubung, dengan menampilkan petunjuk WiFi agar pengguna tahu harus konek ke SSID tertentu.
- Terdapat fitur pengulangan pemeriksaan pesan Telegram, sehingga sistem dapat merespons perintah kapan pun.

Program ini berhasil mengintegrasikan berbagai sensor lingkungan, pengendalian aktuator, dan notifikasi real-time melalui platform Telegram. Sistem sangat cocok untuk deteksi dini kebakaran atau kebocoran gas, dan memberi kontrol serta informasi yang cepat kepada pengguna melalui antarmuka berbasis cloud dan display lokal. Dengan menggunakan ESP8266, sistem menjadi portable, hemat daya, dan murah, menjadikannya solusi tepat untuk monitoring rumah, laboratorium, atau industri skala kecil-menengah.

## LAMPIRAN PROGRAM

```
//
//
//          SISTEM IOT UNTUK MONITORING SUHU, KELEMBABAN
//          SUHU, KELEMBABAN, DAN DETEKSI API
//          MENGGUNAKAN ESP8266 DAN PLATFORM TELEGRAM
//          TEKNOLOGI INFORMASI - FAKULTAS SAINS DAN TEKNOLOGI
//          UIN WALLISONGO SEMARANG
//
//
//
// KELOMPOK: 4
// ANGGOTA:
//      1. Irvan Nurmutakim
//      2. M. Lukmanul Khakim
//      3. Wildan Nur Yusufi
//
#include <UniversalTelegramBot.h> // Library untuk menghubungkan ESP8266
dengan Telegram bot
#include <DHT.h> // Library untuk sensor DHT (suhu dan kelembapan)
#include <LiquidCrystal_I2C.h> // Library untuk mengontrol LCD via I2C
#include <ESP8266WiFi.h> // Library untuk koneksi WiFi menggunakan ESP8266

#define DHTPIN D5 // Pin untuk sensor DHT (D5 pada ESP8266)
#define DHTTYPE DHT11 // Jenis sensor DHT yang digunakan (DHT11)

#define MQPIN A0 // Pin analog untuk sensor MQ (untuk deteksi gas)
#define RELAY1_PIN D3 // Pin digital untuk mengontrol Relay 1

DHT dht(DHTPIN, DHTTYPE); // Membuat objek untuk sensor DHT
LiquidCrystal_I2C lcd(0x27, 16, 2); // Alamat I2C dan ukuran LCD 16x2

// Wi-Fi credentials
const char* ssid = "Kelompok4";
const char* password = "walisongo";

// Telegram Bot Token (dari BotFather)
const char* botToken = "7594165892:AAGj6NJrNQTVu3ptu-d5cp1bbUAr-q8QTTk";
```

```

// Chat ID untuk mengirim pesan (dapat diperoleh melalui @userinfobot di
Telegram)
const char* chatID = "931897538";

// WiFiClientSecure untuk koneksi HTTPS
WiFiClientSecure client; // Klien untuk komunikasi aman
UniversalTelegramBot bot(botToken, client); // Membuat objek bot Telegram

void setup() {
    Serial.begin(115200); // Mulai komunikasi serial untuk debugging
    dht.begin(); // Inisialisasi sensor DHT

    // Inisialisasi pin relay
    pinMode(RELAY1_PIN, OUTPUT);

    // Memastikan relay dalam keadaan mati (HIGH = matikan) saat startup
    digitalWrite(RELAY1_PIN, HIGH); // Matikan relay 1 pada awalnya

    // Inisialisasi LCD
    lcd.init(); // Menggunakan init() untuk inisialisasi LCD
    lcd.backlight(); // Mengaktifkan lampu latar LCD

    // Menampilkan pesan pembuka pada LCD
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("MONITORING SUHU,");
    lcd.setCursor(0, 1);
    lcd.print("KELEMBAPAN & API");
    delay(3000);

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("KELOMPOK 4");
    lcd.setCursor(0, 1);
    lcd.print("IOT SK");
    delay(2000);

    lcd.clear();

```

```

lcd.setCursor(0, 0);
lcd.print("IRVAN, WILDAN");
lcd.setCursor(0, 1);
lcd.print("dan LUKMAN");
delay(2000);

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Prodi TI");
lcd.setCursor(0, 1);
lcd.print("UIN Walisongo");
delay(2000);

bool wifiConnected = false;
while (!wifiConnected) {
    WiFi.begin(ssid, password);
    Serial.println("Menghubungkan ke Wi-Fi...");

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Menghubungkan");
    lcd.setCursor(0, 1);
    lcd.print("ke Wifi.....");

    unsigned long startAttemptTime = millis();

    // Coba koneksi maksimal 10 detik
    while (WiFi.status() != WL_CONNECTED && millis() - startAttemptTime <
10000) {
        delay(1000);
        Serial.println("Mencoba koneksi...");
    }

    if (WiFi.status() == WL_CONNECTED) {
        Serial.println("Terhubung ke WiFi");
        client.setInsecure(); // Disable SSL verification
        wifiConnected = true;
    } else {

```



```

Serial.println("Gagal terhubung ke WiFi");

// Tampilkan "Pakai WiFi Berikut"
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Pakai User&Pass");
lcd.setCursor(0, 1);
lcd.print("WIFI Berikut....");
delay(2000);

// Tampilkan "Kelompok 4" dan "Walisongo"
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("User : Kelompok4");
lcd.setCursor(0, 1);
lcd.print("Pass : walisongo");
delay(8000);

// Coba lagi koneksi WiFi (loop while akan mengulang)
}
}

client.setInsecure(); // Nonaktifkan verifikasi SSL (untuk koneksi cepat)
}

void loop() {
    // Membaca suhu dalam Celcius
    float suhu = dht.readTemperature();
    // Membaca kelembapan
    float kelembapan = dht.readHumidity();

    // Membaca nilai dari sensor MQ
    int mqValue = analogRead(MQPIN); // Baca nilai analog dari sensor MQ

    // Cek jika pembacaan gagal dari sensor DHT
    if (isnan(suhu) || isnan(kelembapan)) {
        Serial.println("Gagal membaca sensor DHT");
        lcd.clear();
    }
}

```

```

        lcd.setCursor(0, 0);
        lcd.print("Sensor error");
        return; // Jika gagal, hentikan fungsi loop
    }

    // Menampilkan suhu, kelembapan, dan nilai MQ ke serial monitor untuk
    debugging
    Serial.print("Suhu: ");
    Serial.print(suhu);
    Serial.print(" *C ");
    Serial.print("Kelembapan: ");
    Serial.print(kelembapan);
    Serial.print(" % ");
    Serial.print("MQ Value: ");
    Serial.println(mqValue); // Menampilkan nilai sensor MQ

    // Menampilkan suhu dan kelembapan di LCD
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Suhu: ");
    lcd.print(suhu);
    lcd.print(" *C");

    lcd.setCursor(0, 1);
    lcd.print("Kelembapan: ");
    lcd.print(kelembapan);
    lcd.print(" %");

    // Kontrol relay berdasarkan nilai sensor MQ
    if (mqValue > 600) { // Jika nilai sensor MQ lebih dari 600, deteksi
    api/gas berbahaya
        digitalWrite(RELAY1_PIN, LOW); // Aktifkan relay (pompa air)
        Serial.println("Relay 1 aktif - Deteksi Api!");

        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("TERDETEKSI API!!!");
        lcd.setCursor(0, 1);

```

```

    lcd.print("Pompa Air Aktif");

    String message = "Peringatan: API terdeteksi! Pompa air aktif.";
    bot.sendMessage(chatID, message, "");

    delay(5000); // Tahan tampilan selama 5 detik
}
else if (mqValue > 400) { // Nilai MQ antara 401 - 600 → indikasi suhu/gas
    ringan
    digitalWrite(RELAY1_PIN, HIGH); // Jangan aktifkan relay
    Serial.println("Deteksi gas ringan - relay tidak aktif");

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Gas Terdeteksi!!");
    lcd.setCursor(0, 1);
    lcd.print("Pantau kondisi");

    String message = "Peringatan: Terdeteksi GAS Berbahaya!!!";
    bot.sendMessage(chatID, message, "");

    delay(5000);
}
else {
    // MQ normal (≤400) → relay tetap mati
    digitalWrite(RELAY1_PIN, HIGH); // Matikan relay
    Serial.println("MQ normal - Relay 1 mati");
}

// Cek jika ada pesan baru dari Telegram
int numNewMessages = bot.getUpdates(bot.last_message_received + 1); //
Mulai dari pesan terbaru
while (numNewMessages) {
    Serial.println("Membaca pesan baru...");

    // Iterasi melalui pesan baru
    for (int i = 0; i < numNewMessages; i++) {
        String text = bot.messages[i].text; // Pesan yang diterima
    }
}

```

```

String fromName = bot.messages[i].from_name; // Nama pengirim

// Jika pesan berisi "cek suhu", kirimkan suhu
if (text == "Cek suhu") {
    String reply = "Suhu saat ini: " + String(suhu) + " *C";
    bot.sendMessage(chatID, reply, "");
}
// Jika pesan berisi "cek kelembapan", kirimkan kelembapan
else if (text == "Cek kelembapan") {
    String reply = "Kelembapan saat ini: " + String(kelembapan) + " %";
    bot.sendMessage(chatID, reply, "");
}
// Jika pesan berisi "cek MQ", kirimkan nilai sensor MQ
else if (text == "Cek MQ") {
    String reply = "Nilai sensor MQ: " + String(mqValue);
    bot.sendMessage(chatID, reply, "");
}
// Jika pesan berisi "cek semua", kirimkan suhu, kelembapan, dan nilai
sensor MQ
else if (text == "Cek semua") {
    String reply = "Suhu: " + String(suhu) + " *C\nKelembapan: " +
String(kelembapan) + " %\nMQ Value: " + String(mqValue);
    bot.sendMessage(chatID, reply, "");
}
}

numNewMessages = bot.getUpdates(bot.last_message_received + 1); //
Periksa pembaruan pesan lagi
}

// Delay agar pembacaan tidak terlalu cepat
delay(2000); // Update setiap 2 detik
Serial.println("Menerima pesan...");
}

```