**MEMENUHI TUGAS PROYEK MINI UTS**

**SISTEM MIKROKONTROLER**

Dosen : Bapak Muchamad Rusdan, S.T., M.T.



Kelompok: 8

Ketua: Aprizal (21552011096)

Anggota:

1: Kurniawan (21552011351)

2: Devis Pringga (23552012004)

Kelas : TIF RM 22 CNS

**TEKNIK INFORMATIKA**

**UNIVERSITAS TEKNOLOGI BANDUNG**

**2025**

# Pemantauan Suhu Kamar Jenazah Berbasis IoT

## 1. Latar Belakang

Ruang pendingin (cold storage) memiliki peranan penting dalam menjaga kualitas dan kesegaran produk yang mudah rusak, seperti bahan makanan, obat-obatan, dan juga jenazah pada kamar Jenazah. Fluktuasi suhu yang tidak terpantau dapat menimbulkan kerusakan, kerugian finansial, serta risiko terhadap kesehatan masyarakat. Oleh karena itu, dibutuhkan sistem pemantauan suhu yang mampu bekerja secara real-time, akurat, serta dapat memberikan notifikasi peringatan saat terjadi anomali suhu.

## 2. Tujuan

* Membangun sistem pemantauan suhu ruang pendingin secara real-time.
* Memberikan peringatan otomatis apabila suhu melebihi atau kurang dari ambang batas yang telah ditentukan.
* Menyediakan visualisasi data suhu melalui dashboard atau aplikasi mobile.

## 3. Komponen Sistem

* Sensor Suhu: DHT22, untuk membaca suhu secara akurat dan berkala.
* Mikrokontroler: ESP32, sebagai pengendali utama untuk membaca data sensor dan mengirimkan ke server.
* Koneksi Internet: Wi-Fi pada ESP32 digunakan untuk konektivitas data ke platform berbasis web.
* Platform IoT: Server dengan PHP dan MySQL untuk menyimpan, mengelola, dan menampilkan data.
* Notifikasi: Pengiriman pesan melalui WhatsApp jika suhu melebihi 8°C, menggunakan Twilio API.

**4. Struktur dan Logika Program** yang digunakan dalam proyek ini, yang melibatkan penggunaan **ESP32, DHT22** (sensor suhu dan kelembapan), **LED, buzzer**, dan pengiriman data suhu ke webhook:

### 1. ****Inisialisasi Perangkat dan Library****

* **Library yang digunakan**:
  + WiFi.h: Untuk menghubungkan ESP32 ke jaringan Wi-Fi.
  + HTTPClient.h: Untuk mengirimkan HTTP request (GET) ke webhook.
  + Wire.h dan LiquidCrystal\_I2C.h: Untuk mengontrol LCD menggunakan I2C.
  + DHTesp.h: Untuk mengontrol sensor suhu dan kelembapan DHT22.
* **Definisi Pin**:
  + DHT\_PIN (GPIO15): Pin untuk menghubungkan sensor DHT22.
  + LED\_MERAH (GPIO2) dan LED\_HIJAU (GPIO27): Pin untuk mengontrol LED (merah untuk suhu tinggi, hijau untuk suhu normal).
  + BUZZER\_PIN (GPIO14): Pin untuk mengontrol buzzer.
* **Variabel Global**:
  + webhookUrl: URL endpoint untuk mengirimkan data suhu ke server.
  + dht: Objek dari kelas DHTesp untuk membaca data dari sensor DHT22.
  + lcd: Objek untuk mengontrol LCD dengan I2C (alamat 0x27).
  + sudahKirim: Variabel boolean untuk memastikan bahwa data suhu hanya dikirimkan satu kali.

### 2. ****Fungsi**** setup()

* **Inisialisasi Serial Monitor**: Serial.begin(115200); untuk komunikasi serial debugging.
* **Inisialisasi DHT22**: dht.setup(DHT\_PIN, DHTesp::DHT22); untuk mengonfigurasi sensor DHT22.
* **Inisialisasi LCD**: lcd.init(); untuk menyalakan LCD dan lcd.backlight(); untuk menyalakan lampu latar LCD.
* **Konfigurasi Pin**:
  + pinMode(LED\_MERAH, OUTPUT); dan pinMode(LED\_HIJAU, OUTPUT); untuk mengatur LED sebagai output.
  + pinMode(BUZZER\_PIN, OUTPUT); untuk mengatur buzzer sebagai output.
* **Koneksi ke Wi-Fi**: WiFi.begin(ssid, password); untuk menghubungkan ESP32 ke jaringan Wi-Fi.

### 3. ****Fungsi**** loop()

* **Pembacaan Data dari DHT22**:
  + TempAndHumidity data = dht.getTempAndHumidity(); untuk mendapatkan suhu dan kelembaban dari sensor DHT22.
  + float suhu = data.temperature; dan float hum = data.humidity; untuk menyimpan data suhu dan kelembaban dalam variabel.
  + **Jika pembacaan suhu gagal**: Program akan menampilkan pesan error di Serial Monitor dan di LCD ("Sensor error").
* **Menampilkan Data di LCD**:
  + lcd.setCursor(0, 0); lcd.print("Suhu: "); lcd.print(suhu, 1); lcd.print(" C"); untuk menampilkan suhu di baris pertama LCD.
  + lcd.setCursor(0, 1); lcd.print("Hum : "); lcd.print(hum, 1); lcd.print(" %"); untuk menampilkan kelembaban di baris kedua LCD.
* **Kondisi Suhu**:
  + **Jika suhu lebih besar dari 8°C**:
    - **LED dan Buzzer**:
      * Nyalakan LED merah (digitalWrite(LED\_MERAH, HIGH);) dan matikan LED hijau (digitalWrite(LED\_HIJAU, LOW);).
      * Nyalakan buzzer (digitalWrite(BUZZER\_PIN, HIGH);), tunggu selama 500ms, kemudian matikan buzzer (digitalWrite(BUZZER\_PIN, LOW);).
    - **Pengiriman Data ke Webhook**:
      * Jika data suhu belum pernah dikirim (!sudahKirim), maka data suhu akan dikirimkan ke server menggunakan HTTP GET request.
      * HTTPClient http; dan http.begin(url); digunakan untuk menginisialisasi dan memulai pengiriman data.
      * Setelah pengiriman data, status sudahKirim diset menjadi true untuk memastikan data hanya dikirim satu kali.
    - **Jika Pengiriman Webhook Berhasil**: Sistem akan menampilkan status pengiriman di Serial Monitor.
  + **Jika suhu lebih kecil atau sama dengan 8°C**:
    - **LED Hijau Menyala**: Nyalakan LED hijau (digitalWrite(LED\_HIJAU, HIGH);) dan matikan LED merah (digitalWrite(LED\_MERAH, LOW);).
    - Set sudahKirim ke false agar sistem siap mengirimkan data lagi jika suhu melebihi 8°C di pembacaan berikutnya.
* **Penundaan**:
  + delay(10000); memberikan jeda 10 detik sebelum pembacaan suhu berikutnya.

### 4. ****Struktur Program****

* **Setup**: Menyiapkan perangkat keras (sensor, LED, buzzer, LCD) dan menghubungkan ke Wi-Fi.
* **Loop**: Pembacaan suhu dan kelembaban, menampilkan data pada LCD, mengontrol LED dan buzzer, serta mengirimkan data ke server jika suhu melebihi ambang batas.

### 5. ****Logika Utama Program****

* Sensor suhu dan kelembaban (DHT22) digunakan untuk membaca suhu dan kelembaban secara berkala.
* Ketika suhu lebih tinggi dari 8°C:
  + LED merah menyala sebagai indikator suhu tinggi.
  + Buzzer berbunyi untuk memberi peringatan.
  + Data suhu dikirimkan ke server menggunakan HTTP GET request ke webhook.
* Ketika suhu normal (≤ 8°C):
  + LED hijau menyala untuk menunjukkan suhu aman.
* Program menghindari pengiriman data yang berulang kali jika suhu tidak berubah.

### 6. ****Ringkasan Logika****

* **Pembacaan suhu** dilakukan setiap 10 detik.
* **Suhu > 8°C**:
  + Nyalakan LED merah, matikan LED hijau.
  + Aktifkan buzzer.
  + Kirim data ke webhook jika belum terkirim.
* **Suhu ≤ 8°C**:
  + Nyalakan LED hijau, matikan LED merah.
* **LCD** selalu menampilkan suhu dan kelembaban terbaru.

## 5. Alur Kerja Sistem

1. Sensor membaca suhu secara berkala (contoh: setiap 1 menit).
2. Mikrokontroler memproses dan mengirim data ke server.
3. Server menyimpan data suhu dan menyajikan grafik tren suhu.
4. Bila suhu melebihi ambang batas, sistem otomatis mengirimkan notifikasi WhatsApp ke pengguna.
5. Pengguna dapat memantau suhu melalui dashboard web atau aplikasi mobile.

## 6. Fitur

* + Monitoring suhu 24/7.
  + Riwayat suhu dan grafik visual.
  + Dukungan multi-ruang pendingin.
  + Notifikasi suhu tinggi melalui WhatsApp.
  + Antarmuka dashboard yang responsif untuk web dan mobile.

## 7. Manfaat

* + Mengurangi risiko kerusakan akibat suhu yang tidak stabil.
  + Meningkatkan efisiensi dan keamanan dalam pengelolaan ruang pendingin.
  + Menjamin kualitas penyimpanan barang sensitif seperti jenazah atau produk farmasi.

## Diagram Alur Sistem

Internet/Cloud (Firebase/MQTT)

Mikrokontroler (ESP32)

Sensor Suhu (DHT22)

Database & Server (Web API) (Firebase Realtime DB / MySQL + PHP API)

Web Dashboard (Grafik, Log)

Mobile App (Notifikasi, Log)

Notifikasi (Email / WhatsApp)

## Simulasi Awal

Tujuan:

• Membaca data suhu dari sensor DHT22.

• Menampilkan data suhu di Serial Monitor.

• Mengaktifkan LED dan buzzer jika suhu > 8°C.

• Mengirimkan notifikasi WhatsApp sebagai alarm.

**Komponen:**

1. ESP32
2. Sensor DHT22
3. LCD 16x2 (l2C)
4. LED + resistor 220Ω
5. Buzzer
6. Bread Board
7. Server /PHP/MySQL
8. Twilio Console/Webhook

## Penjelasan Komponen

1. Sensor DHT22 Berfungsi untuk mengukur suhu dan kelembaban secara berkala di ruang pendingin.
2. ESP32 Menerima data dari sensor dan mengirimkan ke server melalui koneksi Wi-Fi.
3. LCD 16x2 (l2C)Modul tampilan yang bisa menampilkan **2 baris teks, masing-masing 16 karakter**, dan terhubung dengan **antarmuka I2C** sehingga hanya memerlukan **2 pin data (SDA dan SCL).**
4. LED + resistor 220Ω Komponen yang memancarkan cahaya saat dialiri arus dari anoda (+) ke katoda (–), sedangkan resistor untuk **membatasi arus listrik** yang mengalir ke LED agar tidak rusak.
5. Buzzer untuk **menghasilkan suara (bunyi beep)** saat diberi sinyal listrik dari mikrokontroler seperti ESP32 atau Arduino.
6. Bread Board Menyusun dan menghubungkan komponen/pin dari mikrokontroler, **Mengorganisir rangkaian** agar terlihat lebih rapi dan mudah dipahami.
7. Server / Cloud Dapat menggunakan Firebase atau server PHP/MySQL lokal untuk penyimpanan dan pengelolaan data.
8. Twilio Console/Webhook Notifikasi WhatsApp dikirim jika suhu melebihi 8°C, melalui webhook dan Twilio API.

**Hasil Simulasi:**

* Serial monitor menampilkan suhu setiap 2 detik.
* Suhu > 8°C → LED menyala & buzzer aktif.
* Simulasi trigger untuk pengiriman notifikasi WhatsApp berhasil dijalankan via webhook.

## Alur Pengiriman Notifikasi WhatsApp

Twilio API for WhatsApp:

ESP32

HTTP POST

API Request

Webhook PHP

Twilio API

WhatsApp User

**Isi Notifikasi:**

🚨 ALERT: Suhu tinggi terdeteksi!  
📍 Lokasi: Kamar Jenazah lt. 2 Gedung Melati  
🌡️ Suhu Saat Ini: 10.4°C  
🕒 Waktu: 22 April 2025 - 13:30  
⚠️ Harap periksa segera!

**Hasil Implementasi/Link Source Code**

Youtube: <https://youtu.be/j8A5pVdDCMo>

GitHub: <https://github.com/devispringga/kelompok-8.git>

Wokwi: <https://wokwi.com/projects/429201032127397889>

**Pembagian Tugas:**

**Aprizal (21552011096)**: Membuat kode untuk membaca data LED dan menyusun laporan.

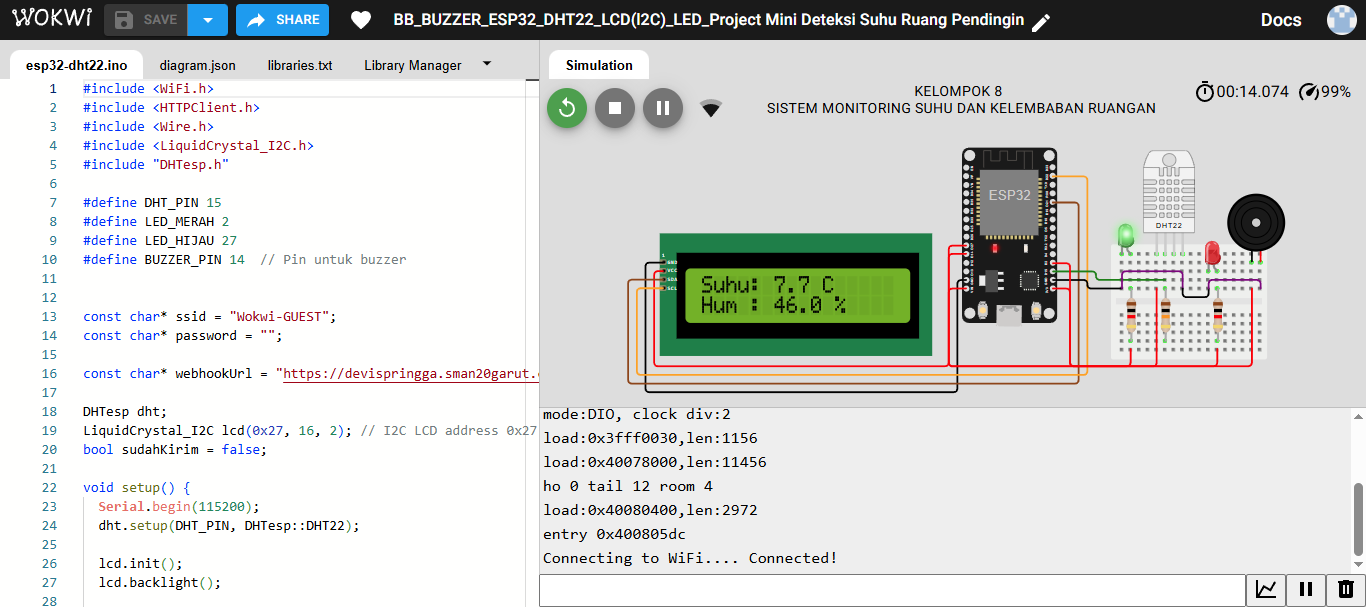
**Kurniawan (21552011351)**: Menyusun/membuat presentasi dan dokumentasi video.

**Devis Pringga (23552012004)**: Konfigurasi/membuat kode untuk membaca data, komponen dan integrasi dengan webhook/twilio dan whatsapp.

**Gambar Konfigurasi Program**

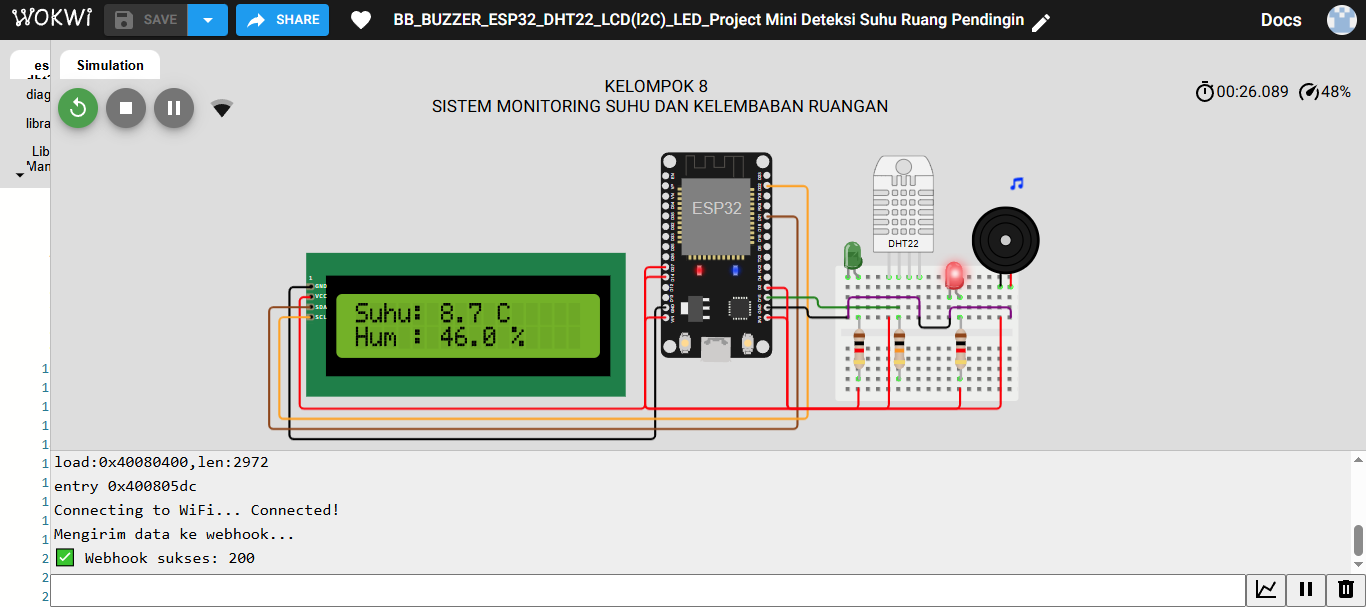
* 1. Halaman Utama Wokwi

Suhu < 8 led hijau menyala tanpa mengirim notifikasi ke WhatsApp



* 1. Halaman Rangkaian Komponen

Suhu > 8 led merah menyala otomatis mengirim notifikasi ke WhatsApp



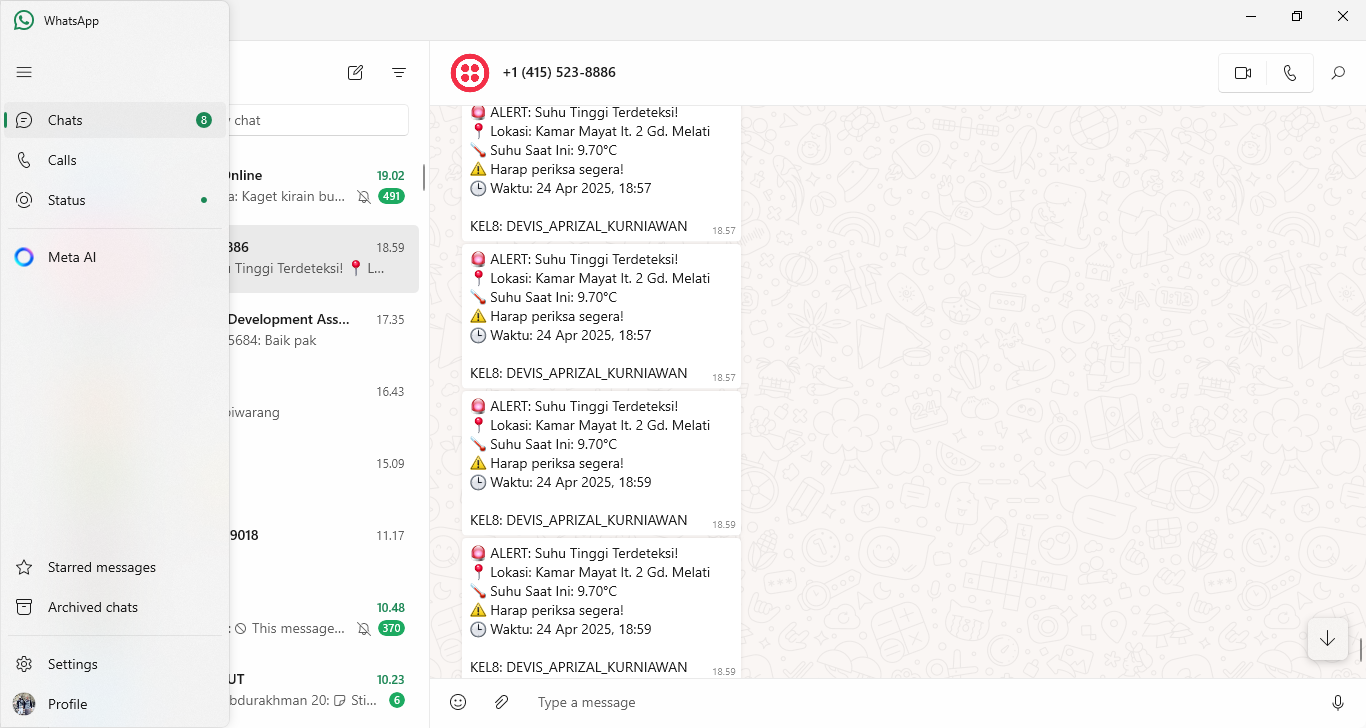
* 1. Source Code



* 1. Konfigurasi Webhook



* 1. Notifikasi WhatsApp



**Refleksi terhadap proses pembuatan proyek mini, tantangan terbesar dihadapi serta solusinya.**

Terdapat banyak tipe ESP32 sehingga mendapat kesulitan merangkai dalam menyesuaikan kode. Untuk solusi harus lebih banyak mencari referensi.

*Terima Kasih*