

**LAPORAN UJIAN AKHIR SEMESTER  
IMPLEMENTASI SISTEM MONITORING ASAP, SUHU, DAN KELEMBAPAN DI  
RUANG SERVER BERBASIS INTERNET OF THINGS**

**Mata Kuliah Internet of Things**



**Dosen Pengampu : Ir. Subairi, ST., MT., IPM**

**Nama Kelompok :**

Ranindya Dwi Qintari	233140707111112
Nova Nizar	233140707111113
Salman Alfarizi Ardiansyah	233140707111124
Rumiris Butarbutar	233140707111130

**Kelas: T4J**

**PROGRAM STUDI TEKNOLOGI INFORMASI**

**FAKULTAS VOKASI**

**UNIVERSITAS BRAWIJAYA**

**2025**

## DAFTAR ISI

DAFTAR ISI .....	2
Abstrak .....	3
BAB 1 PENDAHULUAN .....	4
1.1 Latar Belakang .....	4
1.2 Rumusan Masalah .....	4
1.3 Tujuan .....	4
1.4 Manfaat .....	4
BAB 2 METODOLOGI .....	5
2.1 Alat dan Bahan .....	5
2.2 Langkah Implementasi.....	5
BAB 3 HASIL DAN PEMBAHASAN.....	7
3.1 Perancangan Sistem .....	7
3.1.1 Arsitektur Sistem .....	7
3.1.2 Perancangan Perangkat Keras .....	7
3.1.3 Perancangan Perangkat Lunak (Firmware ESP32) .....	8
3.1.4 Perancangan Antarmuka Website .....	9
3.2 Implementasi dan Pengujian Sistem .....	9
3.2.1 Implementasi Perangkat Keras .....	9
3.2.2 Implementasi Perangkat Lunak (Firmware ESP32) .....	9
3.2.3 Implementasi Website .....	9
3.2.4 Hasil Pengujian Fungsional Sistem (Perangkat Keras).....	9
3.2.5 Hasil Pengujian Integrasi Website .....	11
3.3 Pembahasan .....	12
3.3.1 Simulasi Sistem pada Wokwi dan Vscod​​e .....	12
3.3.2 Analisis Keterbatasan Sistem.....	13
3.3.3 Rekomendasi Pengembangan Lebih Lanjut.....	13
BAB 4 PENUTUP .....	14
4.1 Kesimpulan .....	14
4.2 Saran.....	14
DAFTAR PUSTAKA .....	15
LAMPIRAN .....	16

## Abstrak

Ruang server adalah area krusial yang memerlukan kestabilan lingkungan optimal untuk mencegah kerusakan perangkat keras dan gangguan operasional sistem. Suhu berlebih dan potensi kebakaran akibat asap merupakan ancaman utama yang membutuhkan pemantauan kontinyu. Penelitian ini merancang dan mengimplementasikan sistem pemantauan suhu, kelembapan, dan asap secara *real-time* berbasis *Internet of Things (IoT)*. Sistem ini menggunakan mikrokontroler ESP32 yang terintegrasi dengan sensor DHT22 untuk suhu dan kelembapan, serta sensor MQ-2 untuk deteksi asap. Sistem ini terhubung dengan website yang dibangun menggunakan framework Laravel melalui koneksi Wi-Fi untuk menampilkan data sensor secara langsung di perangkat pengguna. Melalui pengujian perangkat keras, sistem ini terbukti mampu membaca, mengirimkan, dan menampilkan data suhu, kelembapan, serta deteksi asap secara akurat dan *real-time*, memberikan solusi pemantauan dini yang efektif untuk ruang server.

Kata kunci: IoT, ESP32, DHT22, MQ-2, Laravel

*Abstract - Server rooms are crucial areas that require optimal environmental stability to prevent hardware damage and system operational disruptions. Excessive temperatures and potential fires due to smoke are major threats that require continuous monitoring. This study designs and implements a real-time temperature, humidity, and smoke monitoring system based on the Internet of Things (IoT). This system uses an ESP32 microcontroller integrated with a DHT22 sensor for temperature and humidity, and an MQ-2 sensor for smoke detection. This system is connected to a website built using the Laravel framework via a Wi-Fi connection to display sensor data directly on the user's device. Through hardware testing, this system is proven to be able to read, transmit, and display temperature, humidity, and smoke detection data accurately and in real time, providing an effective early monitoring solution for server rooms.*

*Keywords: IoT, ESP32, DHT22, MQ-2, Laravel*

## **BAB 1**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Ruang server memegang peran utama dalam operasional teknologi informasi sebagai tempat pengolahan dan penyimpanan data. Kerusakan pada server dapat mengakibatkan gangguan sistem, kehilangan data, bahkan kegagalan layanan. Faktor-faktor lingkungan seperti suhu berlebih dan kebocoran asap sering menjadi penyebab utama kerusakan perangkat di ruang server.

Dengan kemajuan teknologi Internet of Things (IoT), kini dimungkinkan untuk membangun sistem pemantauan lingkungan secara otomatis dan real-time. Dengan menggunakan mikrokontroler ESP32 serta sensor DHT22 dan MQ-2, sistem dapat mendeteksi suhu, kelembapan, dan asap secara langsung dan mengirimkan datanya ke pengguna melalui platform website. Implementasi ini tidak lagi berbasis simulasi, melainkan menggunakan perangkat keras asli untuk pengujian di dunia nyata.

#### **1.2 Rumusan Masalah**

Rumusan masalah dalam praktik ini adalah:

1. Bagaimana merancang sistem monitoring suhu, kelembapan, dan asap berbasis IoT menggunakan perangkat fisik?
2. Bagaimana sistem mengirimkan data ke aplikasi Blynk secara real-time?

#### **1.3 Tujuan**

Adapun tujuan dari praktik ini adalah:

1. Membangun sistem monitoring suhu, kelembapan, dan asap berbasis ESP32, DHT22, dan MQ-2.
2. Mengintegrasikan sistem monitoring dengan platform website berbasis Laravel agar data dapat dipantau dari jarak jauh secara *real-time*.

#### **1.4 Manfaat**

1. Memberikan sistem pemantauan lingkungan yang efektif dan efisien di ruang server.
2. Membantu teknisi dalam mendeteksi potensi gangguan secara dini.
3. Meningkatkan keamanan dan kestabilan perangkat server.

## **BAB 2**

### **METODOLOGI**

#### **2.1 Alat dan Bahan**

- ESP32
- Sensor DHT22
- Sensor MQ-2
- Breadboard dan kabel jumper
- Koneksi Wi-Fi
- Laptop dengan VSCode (Visual Studio Code) terinstal ekstensi PlatformIO
- Led (Merah, Kuning, Hijau)
- OLED display SSD1306

#### **2.2 Langkah Implementasi**

##### **1. Persiapan Perangkat Keras**

Menyiapkan beberapa komponen utama yang diperlukan dalam proyek ini, antara lain:

- ESP32
- Sensor DHT22 untuk membaca suhu dan kelembapan
- Sensor MQ-2 untuk mendeteksi gas/asap
- Tiga buah LED (Merah, Kuning, Hijau) sebagai indikator
- OLED display tipe SSD1306 untuk menampilkan informasi sensor
- Breadboard dan kabel jumper untuk perakitan komponen

Semua komponen dihubungkan melalui breadboard menggunakan kabel jumper dengan memperhatikan koneksi power (3.3V/5V) dan ground (GND).

##### **2. Perakitan Sistem**

Setelah perangkat keras siap, dilakukan proses perakitan dengan langkah-langkah sebagai berikut:

- Sensor DHT22 dihubungkan ke pin GPIO 4 pada ESP32
- Sensor MQ-2 dihubungkan ke pin GPIO 34 sebagai input analog
- LED Merah, Kuning, dan Hijau masing-masing dihubungkan ke pin GPIO 5, 18, dan 19
- OLED SSD1306 dihubungkan melalui komunikasi I2C dengan pin SDA (GPIO 21) dan SCL (GPIO 22)
- Semua jalur power dan ground disambungkan dengan benar untuk menghindari kesalahan rangkaian

##### **3. Pembuatan Program**

Program dibuat menggunakan Arduino IDE atau VSCode dengan PlatformIO (sesuai lingkungan pengembangan yang digunakan) dengan struktur berikut:

- Menggunakan library DHT, WiFi, dan U8g2
- Inisialisasi sensor, LED, dan OLED
- Membaca data suhu dan kelembapan dari DHT22
- Membaca nilai gas dari sensor MQ-2
- Menentukan status sistem berdasarkan ambang batas nilai sensor
- Menampilkan data pada OLED display
- Mengaktifkan LED indikator sesuai status
- Koneksi ke jaringan Wi-Fi
- Pengiriman data ke server atau API website menggunakan MQTT

##### **4. Pengujian Awal**

Setelah kode program selesai ditulis dan dimasukkan ke dalam file main.cpp, dilakukan pengujian awal untuk memastikan:

- Sensor dapat membaca dan menampilkan data dengan benar
- OLED menampilkan data suhu, kelembapan, gas, dan status sistem
- LED menyala sesuai kondisi lingkungan
- ESP32 berhasil terhubung ke Wi-Fi
- Data berhasil dikirim ke server

##### **5. Desain Tampilan Website**

Website dibangun dengan framework Laravel sebagai backend dan frontend menggunakan HTML, CSS, dan JavaScript. Fitur tampilan mencakup:

- a. Dashboard Monitoring
  - Menampilkan data secara real-time:
  - Grafik suhu, kelembapan, dan gas (menggunakan Chart.js)
  - Status sistem: NORMAL, WASPADA, BAHAYA
  - Warna notifikasi status (merah, kuning, hijau)
  - Indikator LED digital dan status sensor
- b. Tabel Riwayat Sensor
  - Menampilkan log data dari ESP32
  - Data mencakup: waktu, suhu, kelembapan, nilai gas, dan status
  - Bisa disortir atau dicari berdasarkan tanggal
- c. Notifikasi Visual
  - Jika status BAHAYA terdeteksi, akan muncul peringatan di layar (alert warna merah)
  - Status KUNING atau HIJAU juga diberi kode warna visual
- d. Responsif
  - Tampilan dapat diakses melalui desktop dan mobile
  - Menggunakan Bootstrap untuk styling

6. Simulasi dan Uji Coba

Simulasi dilakukan untuk menguji keseluruhan alur sistem:

- Perubahan suhu, kelembapan, atau asap disimulasikan
- Data berhasil terkirim ke server
- Website berhasil menerima dan menampilkan data secara real-time
- Tampilan LED digital dan status berubah sesuai kondisi
- Grafik dan tabel histori berjalan dengan baik

## BAB 3

### HASIL DAN PEMBAHASAN

#### 3.1 Perancangan Sistem

Bagian ini menguraikan bagaimana sistem pemantauan suhu, kelembapan, dan asap dirancang secara komprehensif, mulai dari arsitektur umum hingga perancangan detail pada perangkat keras, perangkat lunak (firmware), dan antarmuka website.

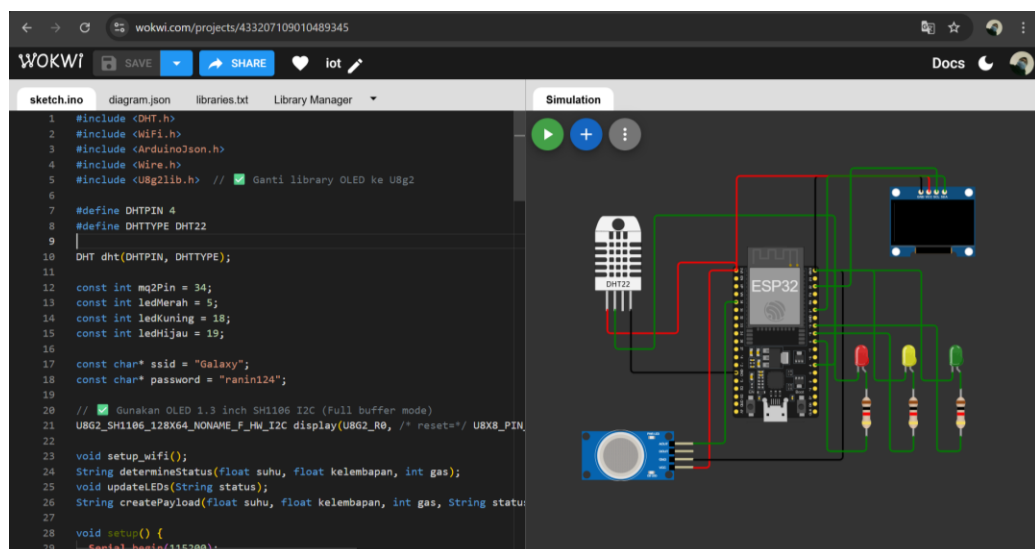
##### 3.1.1 Arsitektur Sistem

Sistem pemantauan ini dirancang dengan arsitektur Internet of Things (IoT) yang memungkinkan pemantauan lingkungan ruang server secara real-time. Komponen utama sistem terdiri dari modul mikrokontroler ESP32 yang terhubung dengan sensor DHT22 untuk suhu dan kelembapan, serta sensor MQ-2 untuk deteksi asap/gas. Data dari sensor diproses oleh ESP32, kemudian dikirimkan secara nirkabel melalui koneksi Wi-Fi menggunakan protokol MQTT menuju server yang mengimplementasikan API berbasis framework Laravel. Data yang diterima oleh server akan disimpan ke dalam database dan ditampilkan melalui antarmuka website.

Alur data dimulai ketika ESP32 secara berkala membaca nilai dari sensor DHT22 dan MQ-2. Data ini kemudian diolah oleh firmware untuk menentukan status lingkungan (NORMAL, WASPADA, BAHAYA) berdasarkan ambang batas yang telah ditetapkan. Selanjutnya, ESP32 mengirimkan data sensor dan status ke server Laravel melalui koneksi Wi-Fi menggunakan protokol MQTT. API Laravel berfungsi sebagai *endpoint* yang menerima data, memprosesnya, dan menyimpannya ke dalam database. Data yang tersimpan di database dapat diakses dan divisualisasikan secara real-time melalui antarmuka website yang responsif.

##### 3.1.2 Perancangan Perangkat Keras

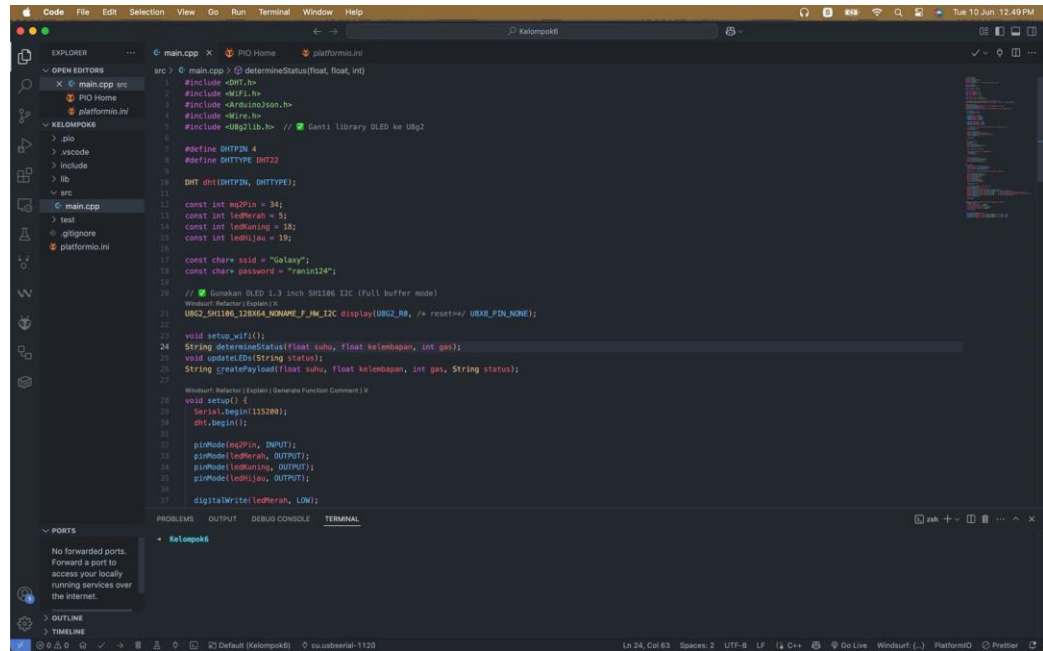
Perancangan perangkat keras mencakup pemilihan dan konektivitas komponen fisik. Perangkat utama yang digunakan adalah mikrokontroler ESP32 sebagai *main controller*. Sensor DHT22 dihubungkan ke pin GPIO 4 pada ESP32 untuk pembacaan data suhu dan kelembapan. Untuk deteksi asap/gas, sensor MQ-2 dihubungkan ke pin GPIO 34 sebagai input analog. Sebagai indikator visual status lingkungan, tiga buah LED (Merah, Kuning, Hijau) masing-masing dihubungkan ke pin GPIO 5, 18, dan 19. Informasi suhu, kelembapan, gas, dan status sistem juga ditampilkan pada OLED display SH1106 yang terhubung melalui komunikasi I2C pada pin SDA (GPIO 21) dan SCL (GPIO 22). Semua komponen dirakit menggunakan *breadboard* dan kabel *jumper*, dengan memperhatikan jalur catu daya (3.3V/5V) dan *ground* (GND) untuk memastikan rangkaian yang benar.



Gambar 3.2: Skematik Rangkaian Sistem pada Wokwi

### 3.1.3 Perancangan Perangkat Lunak (Firmware ESP32)

Perangkat lunak untuk ESP32 dikembangkan menggunakan Visual Studio Code (VSCode) dengan ekstensi PlatformIO, yang mendukung pengembangan firmware untuk ESP32. Pengembangan kode diawali dengan simulasi di platform Wokwi untuk memverifikasi fungsionalitas dasar sensor dan output. Kode ditulis dalam bahasa C++ dengan menggunakan beberapa library esensial seperti DHT.h untuk sensor suhu/kelembapan, WiFi.h untuk konektivitas jaringan, U8g2lib.h untuk kontrol OLED, dan PubSubClient.h untuk komunikasi MQTT.



Gambar 3.3: Pengembangan Kode pada VSCode (PlatformIO)

Logika utama firmware meliputi:

- Inisialisasi: Mengatur pin GPIO, inisialisasi sensor, OLED, dan koneksi Wi-Fi.
- Pembacaan Data Sensor: Membaca nilai suhu dan kelembapan dari DHT22 serta nilai analog dari MQ-2 secara berkala (setiap 2 detik).
- Penentuan Status Sistem: Menentukan status lingkungan berdasarkan ambang batas yang telah ditetapkan:
  - Deteksi Gas: Jika nilai gas dari sensor MQ-2 melebihi ambang batas 1000, LED Merah akan menyala. Kondisi ini mengindikasikan status BAHAYA.
  - Deteksi Suhu: Jika suhu ruangan melebihi 27°C, LED Kuning akan menyala. Kondisi ini mengindikasikan status WASPADA.
  - Deteksi Kelembapan: Jika kelembapan melebihi 80%, LED Hijau akan menyala. Kondisi ini juga mengindikasikan status WASPADA.
  - Kondisi Normal: Jika semua nilai sensor (gas, suhu, dan kelembapan) berada di bawah ambang batas peringatan, tidak ada LED indikator yang menyala. Kondisi ini mengindikasikan status NORMAL.
- Kontrol Indikator Lokal: Mengaktifkan LED (Merah, Kuning, Hijau) dan menampilkan informasi (suhu, kelembapan, gas, status) pada OLED display SH1106 sesuai dengan status yang terdeteksi.
- Koneksi Wi-Fi dan Pengiriman Data: ESP32 secara otomatis akan mencoba terhubung ke jaringan Wi-Fi yang telah dikonfigurasi. Setelah terhubung, data sensor dan status akan dikirimkan secara berkala ke broker MQTT, yang kemudian diteruskan ke API Laravel pada server untuk penyimpanan dan tampilan data.



### 3.1.4 Perancangan Antarmuka Website

Antarmuka website dirancang sebagai *dashboard* pemantauan yang memungkinkan pengguna melihat data secara real-time dari jarak jauh. Website dibangun menggunakan framework Laravel sebagai *backend* untuk mengelola API dan database, serta HTML, CSS, dan JavaScript untuk *frontend*. Untuk kemudahan pengembangan tampilan responsif, digunakan framework Bootstrap. Visualisasi data real-time, khususnya grafik suhu, kelembapan, dan gas, diimplementasikan menggunakan [Chart.js](#).

Fitur utama antarmuka website meliputi:

- Dashboard Monitoring Real-time: Menampilkan nilai suhu, kelembapan, dan gas terbaru, serta status lingkungan terkini (NORMAL, WASPADA, BAHAYA) yang ditunjukkan dengan kode warna visual (hijau, kuning, merah).
- Tabel Riwayat Sensor: Menyajikan log data sensor yang terekam di database, dilengkapi dengan fitur pencarian dan penyaringan (sorting) berdasarkan tanggal atau parameter lainnya.
- Notifikasi Visual: Memberikan peringatan visual di layar (misalnya *alert* berwarna merah) saat status BAHAYA terdeteksi, serta kode warna untuk status WASPADA dan NORMAL.
- Responsivitas: Tampilan website dioptimalkan agar dapat diakses dengan baik dan nyaman melalui berbagai perangkat, baik desktop maupun mobile.

## 3.2 Implementasi dan Pengujian Sistem

Bagian ini menjelaskan proses implementasi sistem berdasarkan perancangan yang telah dibuat, diikuti dengan pemaparan hasil pengujian untuk memverifikasi fungsionalitas dan performa keseluruhan sistem.

### 3.2.1 Implementasi Perangkat Keras

Proses implementasi perangkat keras melibatkan perakitan fisik semua komponen yang telah dirancang. Mikrokontroler ESP32, sensor DHT22, sensor MQ-2, LED indikator, dan OLED display SH1106 dihubungkan pada *breadboard* menggunakan kabel *jumper* sesuai dengan skematik yang telah ditentukan. Semua koneksi daya dan *ground* dipastikan terhubung dengan benar untuk menghindari kesalahan rangkaian.

### 3.2.2 Implementasi Perangkat Lunak (Firmware ESP32)

Firmware yang telah dikembangkan menggunakan VSCode dengan PlatformIO di-*compile* dan di-*upload* ke mikrokontroler ESP32. Proses *upload* dilakukan melalui koneksi USB, memastikan bahwa kode program berhasil tertanam pada *flash memory* ESP32 dan siap untuk dieksekusi.

### 3.2.3 Implementasi Website

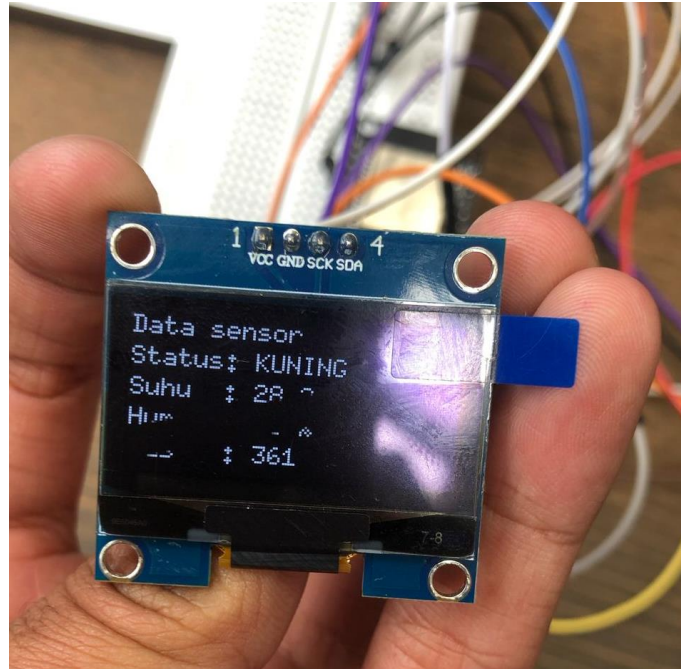
Website pemantauan dibangun dengan framework Laravel dan di-*deploy* pada lingkungan server lokal. Konfigurasi API pada Laravel disiapkan untuk menerima *payload* data sensor yang dikirimkan oleh ESP32 melalui MQTT. Setelah data diterima, API akan menyimpan data ke dalam database yang telah dikonfigurasi. Antarmuka *frontend* yang telah dirancang diintegrasikan dengan *backend* Laravel untuk menampilkan data sensor dan status secara real-time dari database.

### 3.2.4 Hasil Pengujian Fungsional Sistem (Perangkat Keras)

Pengujian fungsional sistem dilakukan secara langsung pada perangkat keras untuk memverifikasi setiap komponen bekerja sesuai fungsinya.

#### 3.2.4.1 Pengujian Pembacaan Sensor dan Tampilan Lokal (OLED & LED)

Setelah ESP32 dinyalakan, sistem berhasil melakukan inisialisasi sensor DHT22 dan MQ-2, serta OLED display. Data suhu, kelembapan, dan gas berhasil dibaca secara berkala (setiap 2 detik) dan ditampilkan secara *real-time* melalui Serial Monitor pada VSCode serta pada OLED display.



Gambar 3.4: Tampilan Data pada OLED Display SH1106

Pengujian status dan indikator LED menunjukkan sistem merespons dengan benar sesuai ambang batas yang ditetapkan:

- Jika nilai gas melebihi 1000, status berubah menjadi BAHAYA dan LED merah menyala
- Jika suhu naik melebihi 27°C, status berubah menjadi WASPADA dan LED kuning menyala.
- Jika kelembapan naik melebihi 80%, status berubah menjadi WASPADA dan LED hijau menyala.
- Ketika semua nilai sensor berada dalam batas normal, status yang ditampilkan adalah NORMAL dan tidak ada LED indikator yang menyala.

#### 3.2.4.2 Pengujian Respon Sensor

Pengujian dilakukan dengan memberikan variasi kondisi lingkungan pada sensor untuk mengamati respons sistem. Suhu ditingkatkan menggunakan korek api gas, kelembapan ditingkatkan dengan ruangan, dan asap diberikan dari rokok.

Berikut adalah hasil uji respons sensor yang menunjukkan kemampuan sistem dalam mendeteksi dan merespons perubahan lingkungan:

No.	Skenario Pengujian	Kondisi Sensor	Status Terdeteksi	LED Menyala	Waktu Respons (detik)	Keterangan
1.	Kondisi Normal	Gas: 150, Suhu: 25°C, Kelembapan: 60%	NORMAL	-	-	Baseline
2.	Peningkatan Suhu	Gas: 155, Suhu: 29°C, Kelembapan: 62%	WASPADA	Kuning	5s	Suhu melebihi ambang batas

3.	Peningkatan Kelembapan	Gas: 153, Suhu: 26°C, Kelembapan: 85%	WASPADA	Hijau	5s	Kelembapan melebihi ambang batas
4.	Deteksi Asap/Gas	Gas: 1100, Suhu: 25°C, Kelembapan: 61%	BAHAYA	Merah	5s	Gas melebihi ambang batas
5.	Kombinasi Asap & Suhu Tinggi	Gas: 1120, Suhu: 30°C, Kelembapan: 65%	BAHAYA	Merah	5s	Prioritas status BAHAYA (Merah tetap aktif)

Tabel 3.1: Hasil Pengujian Respons Sensor

Dari hasil pengujian ini, terlihat bahwa sistem mampu merespons perubahan kondisi lingkungan dengan cepat dan akurat, ditunjukkan oleh perubahan status, aktivasi LED yang sesuai, dan waktu respons yang relatif singkat untuk setiap skenario.

### 3.2.5 Hasil Pengujian Integrasi Website

Pengujian ini berfokus pada kemampuan sistem dalam mengirimkan data ke server dan menampilkan data tersebut melalui antarmuka website Laravel.

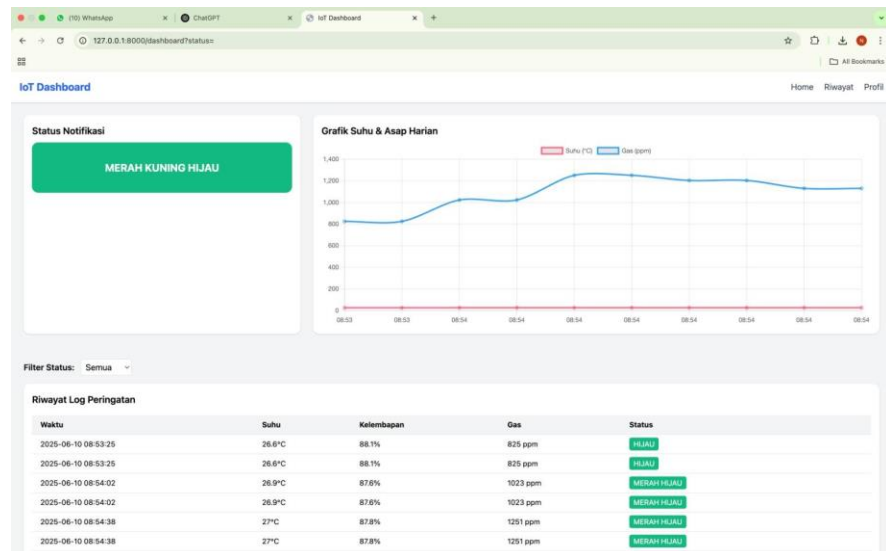
#### 3.2.5.1 Pengujian Konektivitas Wi-Fi dan Pengiriman Data

Pengujian menunjukkan bahwa ESP32 berhasil terhubung ke jaringan Wi-Fi yang telah dikonfigurasi. Konfirmasi koneksi ditunjukkan oleh *output* alamat IP lokal perangkat pada Serial Monitor. Setelah terhubung, ESP32 secara konsisten mengirimkan data sensor dan status ke server Laravel melalui protokol MQTT. Data berhasil diterima oleh API Laravel dan disimpan ke dalam database tanpa kendala signifikan, menunjukkan stabilitas koneksi dan integritas pengiriman data.

#### 3.2.5.2 Pengujian Tampilan Data di Website

Website yang dibangun dengan Laravel berhasil menampilkan data sensor (suhu, kelembapan, gas) dan status lingkungan secara real-time.

- Dashboard Monitoring: Data terbaru diperbarui secara dinamis pada dashboard. Visualisasi data melalui grafik yang diimplementasikan dengan Chart.js berfungsi dengan baik, menunjukkan tren perubahan suhu, kelembapan, dan gas seiring waktu.



Gambar 3.5: Tampilan Dashboard Pemantauan pada Website

- Notifikasi Visual: Perubahan status (NORMAL, WASPADA, BAHAYA) direpresentasikan dengan perubahan warna yang jelas (hijau, kuning, merah) pada antarmuka web, memberikan informasi visual yang intuitif kepada pengguna.
- Notifikasi Visual: Perubahan status (NORMAL, WASPADA, BAHAYA) direpresentasikan dengan perubahan warna yang jelas (hijau, kuning, merah) pada antarmuka web, memberikan informasi visual yang intuitif kepada pengguna.

The screenshot shows the 'Riwayat Log Peringatan' table with the following data:

Waktu	Suhu	Kelembapan	Gas	Status
2025-06-10 08:55:36	27.2°C	85.6%	1023 ppm	MERAH KUNING HIJAU
2025-06-10 08:55:36	27.2°C	85.6%	1023 ppm	MERAH KUNING HIJAU
2025-06-10 08:55:23	27.1°C	86.1%	1067 ppm	MERAH KUNING HIJAU
2025-06-10 08:55:23	27.1°C	86.1%	1067 ppm	MERAH KUNING HIJAU
2025-06-10 08:55:08	27.1°C	85.3%	1103 ppm	MERAH KUNING HIJAU
2025-06-10 08:55:08	27.1°C	85.3%	1103 ppm	MERAH KUNING HIJAU
2025-06-10 08:54:56	27.1°C	85.6%	1130 ppm	MERAH KUNING HIJAU
2025-06-10 08:54:56	27.1°C	85.6%	1130 ppm	MERAH KUNING HIJAU
2025-06-10 08:54:44	27°C	87.1%	1203 ppm	MERAH HIJAU
2025-06-10 08:54:44	27°C	87.1%	1203 ppm	MERAH HIJAU
2025-06-10 08:54:38	27°C	87.8%	1251 ppm	MERAH HIJAU
2025-06-10 08:54:38	27°C	87.8%	1251 ppm	MERAH HIJAU
2025-06-10 08:54:02	26.9°C	87.6%	1023 ppm	MERAH HIJAU
2025-06-10 08:54:02	26.9°C	87.6%	1023 ppm	MERAH HIJAU
2025-06-10 08:53:25	26.6°C	88.1%	825 ppm	HIJAU

Gambar 3.6: Tampilan Tabel Riwayat Data Sensor pada Website

### 3.3 Pembahasan

Bagian pembahasan ini menganalisis performa sistem yang telah diimplementasikan dan diuji, serta mengidentifikasi keterbatasan dan memberikan rekomendasi untuk pengembangan lebih lanjut.

#### 3.3.1 Simulasi Sistem pada Wokwi dan Vscode

Sistem pemantauan suhu, kelembapan, dan gas berbasis ESP32 ini menunjukkan performa yang cukup memuaskan dalam mencapai tujuan penelitian.

- Akurasi dan Stabilitas Sensor: Sensor DHT22 dan MQ-2 mampu membaca nilai lingkungan dengan stabilitas yang baik dan akurasi yang memadai untuk kebutuhan pemantauan dini. Data

yang terekam menunjukkan konsistensi dengan kondisi lingkungan yang diberikan selama pengujian.

- Kecerdasan Logika: Implementasi logika penentuan status (NORMAL, WASPADA, BAHAYA) yang jelas dan berprioritas memungkinkan sistem untuk secara otomatis mengklasifikasikan kondisi lingkungan dan memberikan indikasi yang relevan.
- Visualisasi Lokal: Penggunaan OLED display dan LED indikator terbukti efektif dalam memberikan informasi status lingkungan secara langsung di lokasi perangkat, memudahkan pemantauan cepat tanpa harus membuka antarmuka website.
- Konektivitas dan Integrasi Data: Koneksi Wi-Fi berjalan stabil, memungkinkan pengiriman data yang lancar dari ESP32 ke server Laravel. Integrasi data melalui MQTT dan API Laravel juga berhasil, memastikan data sampai ke database dengan baik dan dapat diakses oleh website secara *real-time*.
- Antarmuka Pengguna (Website): Website yang responsif dengan grafik dan tabel histori memfasilitasi pemantauan jarak jauh yang efisien dan memberikan gambaran menyeluruh tentang kondisi ruang server.

### 3.3.2 Analisis Keterbatasan Sistem

Meskipun menunjukkan performa yang baik, sistem ini memiliki beberapa keterbatasan:

- Waktu Respons Sensor MQ-2: Sensor MQ-2, meskipun efektif untuk deteksi asap/gas, menunjukkan waktu respons yang cenderung lambat untuk konsentrasi gas yang sangat rendah. Hal ini berpotensi menyebabkan sedikit keterlambatan dalam deteksi dini pada fase awal kejadian. Keterbatasan ini umumnya merupakan karakteristik sensor semikonduktor oksida logam yang memerlukan waktu pemanasan dan stabilisasi.
- Absennya Alarm Aktif: Sistem belum dilengkapi dengan fitur alarm aktif (seperti *buzzer* atau notifikasi *push* ke perangkat mobile) untuk kondisi BAHAYA. Ini berarti, meskipun status terdeteksi dan ditampilkan di website, tidak ada peringatan otomatis yang langsung menarik perhatian teknisi jika mereka tidak memantau website secara terus-menerus, yang kurang ideal untuk skenario darurat di ruang server

### 3.3.3 Rekomendasi Pengembangan Lebih Lanjut

Berdasarkan hasil evaluasi dan keterbatasan yang teridentifikasi, beberapa rekomendasi pengembangan lebih lanjut dapat diusulkan:

- Penambahan Sistem Alarm Aktif: Mengintegrasikan *buzzer* pada perangkat fisik ESP32 dan/atau fitur notifikasi *push* ke *smartphone* (misalnya melalui Telegram API atau Firebase Cloud Messaging) untuk memberikan peringatan instan saat terdeteksi kondisi BAHAYA.
- Peningkatan Akurasi dan Kecepatan Deteksi Gas: Melakukan kalibrasi lebih lanjut pada sensor MQ-2 atau mempertimbangkan penggunaan sensor gas yang lebih sensitif dan spesifik untuk asap.
- Penambahan Sensor Lain: Mengembangkan sistem dengan menambahkan sensor-sensor lain yang relevan untuk ruang server, seperti sensor aliran udara untuk memantau ventilasi, sensor kebocoran air, atau sensor pintu.
- Fitur Kontrol Otomatis: Mengimplementasikan fungsi kontrol otomatis, seperti mengaktifkan kipas atau sistem pendingin jika suhu melebihi batas tertentu.
- Analisis Data Historis Lanjutan: Memanfaatkan data historis yang terkumpul untuk melakukan analisis tren atau memprediksi potensi masalah di masa depan melalui algoritma *machine learning* sederhana.

## BAB 4 PENUTUP

### 4.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem pemantauan suhu, kelembapan, dan asap berbasis IoT untuk ruang server, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Sistem pemantauan suhu, kelembapan, dan asap berhasil dirancang dan diimplementasikan menggunakan mikrokontroler ESP32 yang terintegrasi dengan sensor DHT22 untuk suhu dan kelembapan, serta sensor MQ-2 untuk deteksi asap/gas. Perancangan perangkat keras dan perangkat lunak (firmware) telah berhasil dilakukan, memungkinkan pembacaan data sensor secara akurat dan penentuan status lingkungan (NORMAL, WASPADA, BAHAYA) berdasarkan ambang batas yang telah ditetapkan.
2. Sistem mampu mengirimkan data suhu, kelembapan, dan deteksi asap secara real-time dari perangkat ESP32 ke website berbasis framework Laravel melalui koneksi Wi-Fi menggunakan protokol MQTT. Pengujian menunjukkan bahwa data berhasil diterima, disimpan ke database, dan divisualisasikan secara dinamis dalam bentuk grafik dan tabel histori pada antarmuka website yang responsif dan dapat diakses dari berbagai perangkat.
3. Implementasi sistem ini menyediakan solusi pemantauan lingkungan yang efektif dan efisien untuk ruang server, serta membantu teknisi dalam mendeteksi potensi gangguan secara dini dan meningkatkan keamanan serta kestabilan perangkat server. Meskipun demikian, masih terdapat ruang untuk pengembangan lebih lanjut dalam hal sistem notifikasi aktif dan akurasi deteksi gas pada konsentrasi rendah.

### 4.2 Saran

Untuk pengembangan sistem pemantauan suhu, kelembapan, dan asap berbasis IoT di masa mendatang, beberapa saran yang dapat dipertimbangkan adalah:

1. Penambahan Fitur Notifikasi Aktif: Mengimplementasikan sistem peringatan yang lebih proaktif, seperti notifikasi *push* ke perangkat *smartphone* (misalnya melalui API Telegram atau Firebase Cloud Messaging) atau integrasi *buzzer* pada perangkat fisik. Hal ini akan memastikan respons yang lebih cepat terhadap kondisi BAHAYA, bahkan ketika teknisi tidak memantau *dashboard* website secara terus-menerus.
2. Peningkatan Akurasi dan Kecepatan Deteksi Gas: Melakukan kalibrasi lanjutan pada sensor MQ-2 atau mempertimbangkan penggunaan jenis sensor gas lain yang lebih sensitif dan memiliki waktu respons lebih cepat untuk deteksi dini asap/gas dengan konsentrasi sangat rendah.
3. Pengembangan Fitur Kontrol Otomatis: Mengembangkan kemampuan sistem untuk tidak hanya memantau tetapi juga mengontrol perangkat lain di ruang server, seperti mengaktifkan kipas pendingin atau sistem ventilasi secara otomatis jika suhu atau kelembapan melewati batas tertentu.
4. Integrasi Sensor Tambahan: Menambahkan sensor lain yang relevan untuk pemantauan ruang server yang lebih komprehensif, seperti sensor aliran udara, sensor kebocoran air, atau sensor pembukaan pintu untuk keamanan fisik.
5. Analisis Data Lanjutan: Memanfaatkan data historis yang terkumpul untuk melakukan analisis tren atau mengembangkan model prediktif menggunakan teknik *machine learning* sederhana, guna mengidentifikasi pola anomali atau memprediksi potensi masalah sebelum terjadi.

## DAFTAR PUSTAKA

- Aditya, Y., & Hidayat, N. (2021). Rancang bangun sistem monitoring suhu dan kelembaban ruang server berbasis IoT menggunakan protokol MQTT. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 5(10), 4321–4328.
- Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, 17(4), 2347–2376. <https://doi.org/10.1109/COMST.2015.2444095>
- Aosong Electronics Co., Ltd. (2016). *DHT22 (AM2302) digital-output relative humidity & temperature sensor/module* [Datasheet]. Sparkfun. <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>
- Espressif Systems. (2023). *ESP32 series datasheet*. [https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf)
- Hanwei Electronics Co., Ltd. (2018). *Technical data: MQ-2 gas sensor* [Datasheet]. Sparkfun. <https://www.sparkfun.com/datasheets/Sensors/MQ-2.pdf>
- Hillar, G. C. (2018). *Laravel up and running: A framework for building modern PHP apps*. O'Reilly Media.
- Laravel. (2024). *Laravel documentation*. Diakses pada 12 Juni 2025, dari <https://laravel.com/docs>
- OASIS. (2019). *MQTT version 5.0* (OASIS Standard). <https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html>
- PlatformIO. (2024). *PlatformIO: Your gateway to embedded systems*. Diakses pada 12 Juni 2025, dari <https://platformio.org/>
- Wokwi. (2024). *Wokwi - Online simulator for Arduino and ESP32*. Diakses pada 12 Juni 2025, dari <https://wokwi.com/>

## LAMPIRAN

- **Kode Program (Firmware ESP32)**

Bagian ini memuat seluruh kode program (firmware) yang di-*upload* ke mikrokontroler ESP32. Kode ini bertanggung jawab atas pembacaan data sensor, penentuan status lingkungan, serta komunikasi Wi-Fi dan pengiriman data ke server.

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <DHT.h>
#include <ArduinoJson.h>
#include <Wire.h>
#include <U8g2lib.h>

#define DHTPIN 4
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

const int mq2Pin = 34;
const int ledMerah = 5;
const int ledKuning = 18;
const int ledHijau = 19;

const char* ssid = "Galaxy";
const char* password = "ranin124";
const char* mqtt_server = "broker.emqx.io";
const char* mqtt_topic = "ranindya/iot/sensor/data";

WiFiClient espClient;
PubSubClient client(espClient);

U8G2_SH1106_128X64_NONAME_F_HW_I2C display(U8G2_R0, /* reset= */ U8X8_PIN_NONE);

void setup_wifi();
String determineStatus(float suhu, float kelembapan, int gas);
void updateLEDs(String status);
String createPayload(float suhu, float kelembapan, int gas, String status);
void reconnect();

void setup() {
  Serial.begin(115200);
  dht.begin();

  pinMode(mq2Pin, INPUT);
  pinMode(ledMerah, OUTPUT);
  pinMode(ledKuning, OUTPUT);
  pinMode(ledHijau, OUTPUT);

  digitalWrite(ledMerah, LOW);
  digitalWrite(ledKuning, LOW);
  digitalWrite(ledHijau, LOW);

  Wire.begin(21, 22); // SCL, SDA
```



```

display.begin();
display.clearBuffer();
display.setFont(u8g2_font_6x10_tf);
display.drawStr(0, 10, "Monitoring Start");
display.sendBuffer();

setup_wifi();

client.setServer(mqtt_server, 1883);
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();

  float suhu = dht.readTemperature();
  float kelembapan = dht.readHumidity();
  int gasValue = analogRead(mq2Pin);

  if (isnan(suhu) || isnan(kelembapan)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  String status = determineStatus(suhu, kelembapan, gasValue);
  updateLEDs(status);

  String payload = createPayload(suhu, kelembapan, gasValue, status);
  client.publish(mqtt_topic, payload.c_str());

  Serial.println(payload);

  // OLED Display
  display.clearBuffer();
  display.setFont(u8g2_font_6x10_tf);
  display.setCursor(0, 10); display.print("Data Sensor");
  display.setCursor(0, 22); display.print("Status: "); display.print(status);
  display.setCursor(0, 34); display.print("Suhu : "); display.print(suhu, 1); display.print(" C");
  display.setCursor(0, 46); display.print("Hum : "); display.print((int)kelembapan); display.print(" %");
  display.setCursor(0, 58); display.print("Gas : "); display.print(gasValue);
  display.sendBuffer();

  delay(5000); // jeda 5 detik
}

void setup_wifi() {
  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

```

```

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

Serial.println();
Serial.println("WiFi connected");
Serial.print("IP address: ");
Serial.println(WiFi.localIP());
}

void reconnect() {
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");
        if (client.connect("ESP32Client")) {
            Serial.println("connected");
        } else {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" try again in 5 seconds");
            delay(5000);
        }
    }
}

String determineStatus(float suhu, float kelembapan, int gas) {
    String status = "";
    if (gas > 1000) status += "MERAH ";
    if (suhu > 27) status += "KUNING ";
    if (kelembapan > 80) status += "HIJAU ";
    if (status == "") status = "NORMAL";
    return status;
}

void updateLEDs(String status) {
    digitalWrite(ledMerah, status.indexOf("MERAH") >= 0 ? HIGH : LOW);
    digitalWrite(ledKuning, status.indexOf("KUNING") >= 0 ? HIGH : LOW);
    digitalWrite(ledHijau, status.indexOf("HIJAU") >= 0 ? HIGH : LOW);
}

String createPayload(float suhu, float kelembapan, int gas, String status) {
    StaticJsonDocument<256> doc;
    doc["suhu"] = suhu;
    doc["kelembapan"] = kelembapan;
    doc["gas"] = gas;
    doc["status"] = status;

    String payload;
    serializeJson(doc, payload);
    return payload;
}

```

- **Struktur Database Website (Laravel)**

Bagian ini menampilkan skema atau struktur tabel database yang digunakan oleh *backend* Laravel untuk menyimpan data sensor yang diterima dari ESP32

uas_iot logs	
id	bigint(20) unsigned
suhu	double
kelembapan	double
gas	double
status	varchar(255)
created_at	timestamp
updated_at	timestamp

uas_iot users	
id	bigint(20) unsigned
name	varchar(255)
email	varchar(255)
email_verified_at	timestamp
password	varchar(255)
remember_token	varchar(100)
created_at	timestamp
updated_at	timestamp

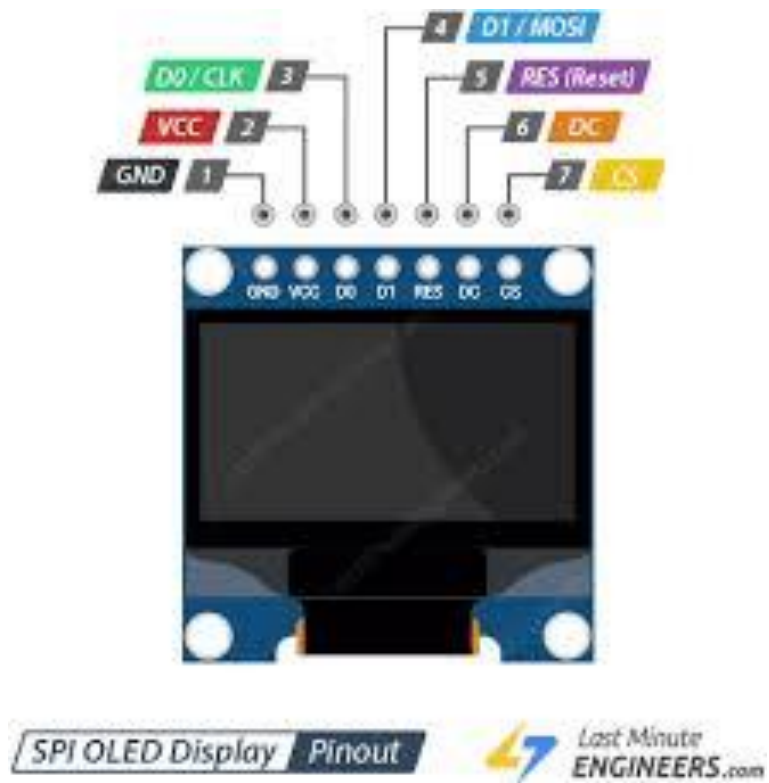
- **Datasheet Komponen Utama**

Bagian ini berisi ringkasan atau bagian relevan dari *datasheet* komponen elektronik utama yang digunakan dalam proyek ini.

- Pinout ESP32

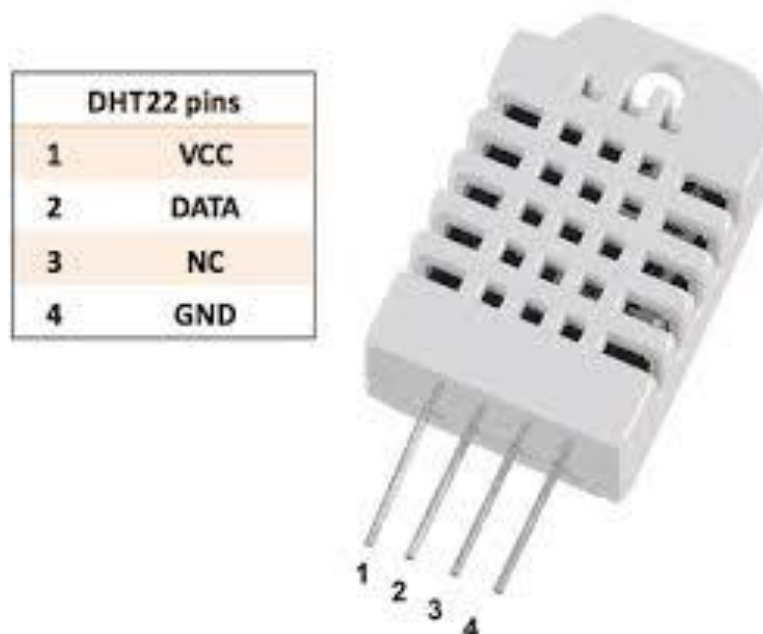


- Pinout OLED 1306



Gambar 5.2: Pinout OLED Display

- Pinout DHT22



Gambar 5.3: Pinout DHT22

- Pinout MQ-2

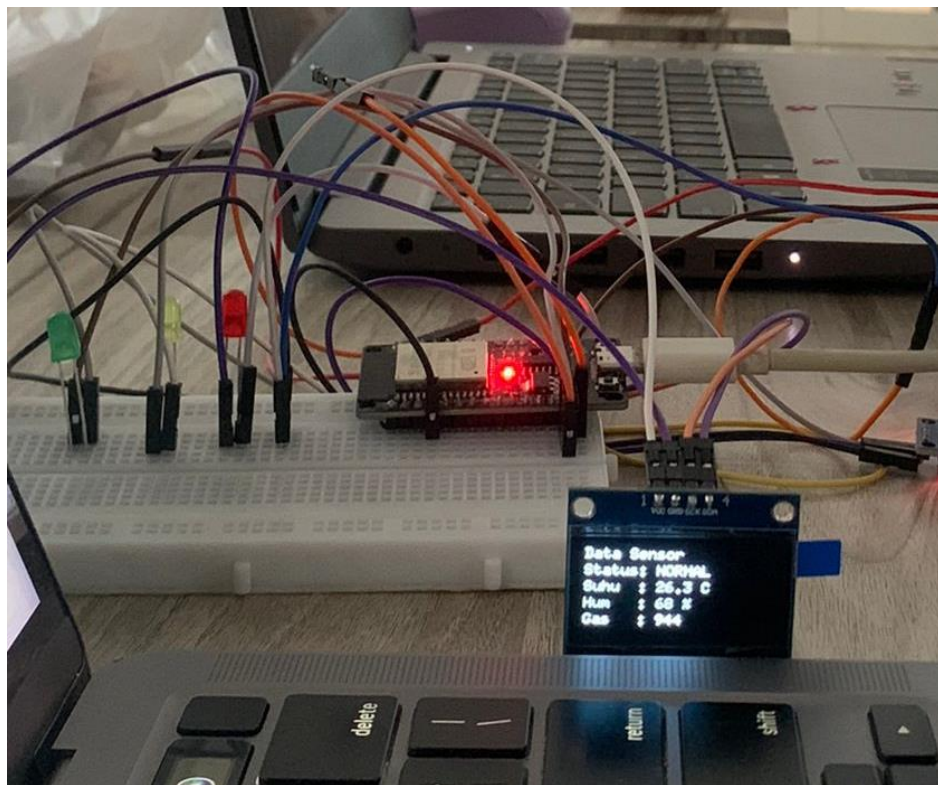


Gambar 5.4: Pinout MQ2

- **Dokumentasi Foto Pengujian Deteksi**

Bagian ini menyajikan dokumentasi visual berupa foto-foto yang diambil saat sistem mendeteksi berbagai kondisi lingkungan dan menunjukkan respons LED indikator yang sesuai. Foto-foto ini menjadi bukti langsung dari fungsionalitas sistem.

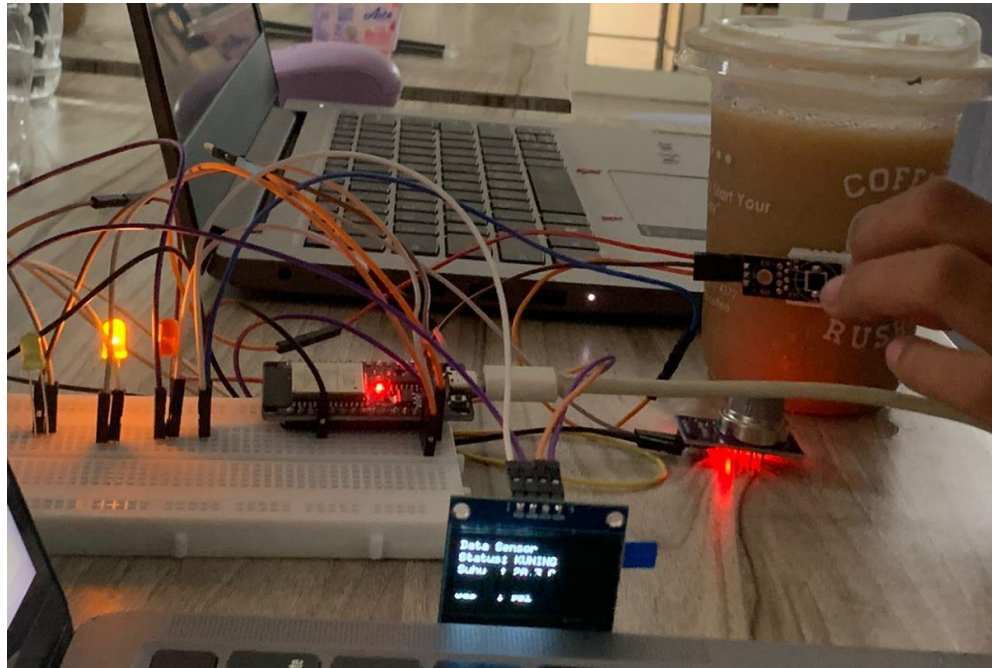
- Foto sistem saat semua sensor dalam batas normal, menunjukkan tidak ada LED yang menyala.



Gambar 5.5: Kondisi Normal (LED Tidak Menyala)

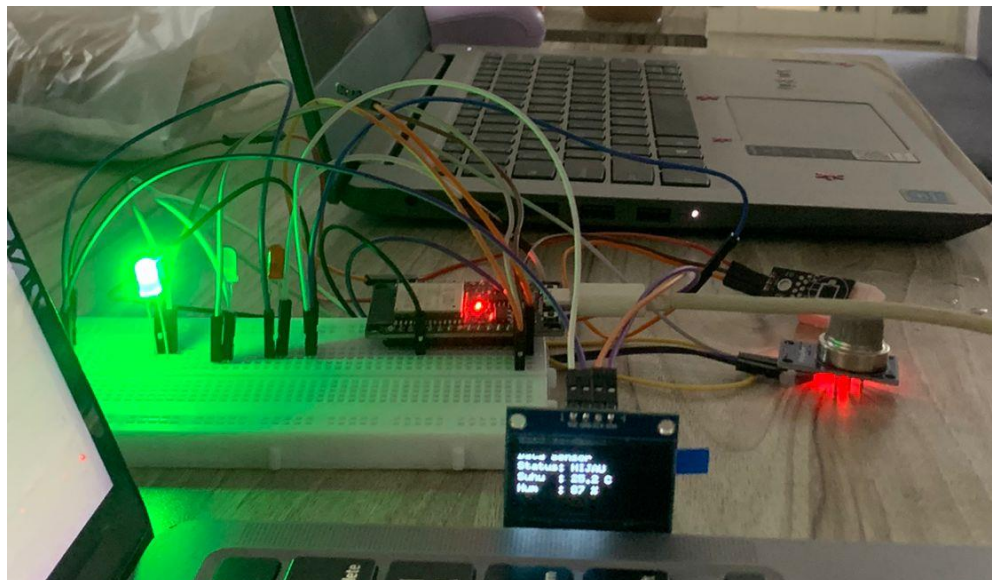


- Foto sistem saat suhu dingin ditingkatkan dengan menggunakan air dingin atau air es, menunjukkan LED kuning menyala dan tampilan OLED yang relevan.



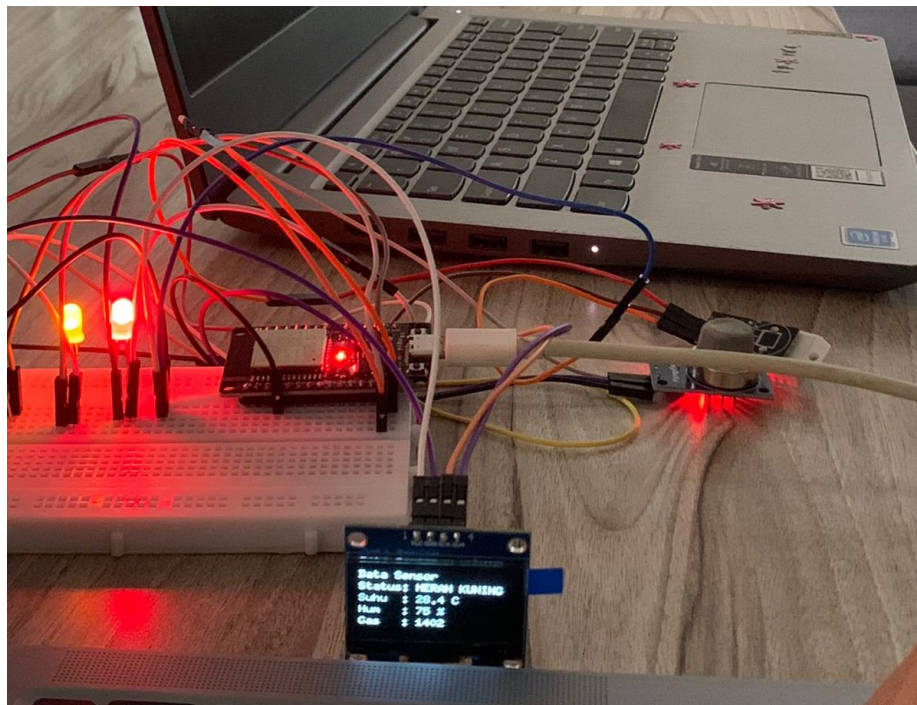
Gambar 5.6: Deteksi Suhu Dingin Tinggi (LED Kuning Menyala)

- Foto sistem saat melakukan deteksi kelembapan ruangan, menunjukkan LED hijau menyala dan tampilan OLED yang relevan.



Gambar 5.7: Deteksi Kelembapan Tinggi (LED Hijau Menyala)

- Foto sistem saat asap/gas didekatkan ke sensor, menunjukkan LED merah menyala dan tampilan OLED yang relevan.



Gambar 5.8: Deteksi Asap/Gas (LED Merah Menyala)

- Foto sistem saat sensor deteksi kelembapan ruangan, suhu dingin ditingkatkan dengan menggunakan air dingin atau air es, asap/gas didekatkan ke sensor, menunjukkan semua LED menyala dan tampilan OLED yang relevan.



Gambar 5.9: Deteksi Kombinasi Asap & Suhu Tinggi ( Semua LED Menyala)