

**SISTEM *MONITORING* KADAR GAS KARBON DIOKSIDA  
(CO<sub>2</sub>) BERBASIS *INTERNET OF THINGS***

**Skripsi  
untuk memenuhi sebagian persyaratan  
mencapai derajat Sarjana S-1 Program Studi  
Teknik Komputer**



**M. ARGHA WIRAYUDHA  
1903010021**

**PROGRAM STUDI TEKNIK KOMPUTER  
FAKULTAS SAINS TEKNOLOGI DAN PENDIDIKAN  
UNIVERSITAS MUHAMMADIYAH LAMONGAN  
2023**

## HALAMAN PENGESAHAN

Skripsi dengan judul :

### **SISTEM MONITORING KADAR GAS KARBON DIOKSIDA (CO<sub>2</sub>) BERBASIS INTERNET OF THINGS**

Oleh:

**M. ARGHA WIRAYUDHA**

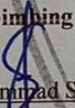
1903010021

Telah dilakukan pembimbingan Skripsi dan dinyatakan layak untuk mengikuti ujian skripsi pada Program Studi Teknik Komputer Fakultas Sains, Teknologi, dan Pendidikan Universitas Muhammadiyah Lamongan

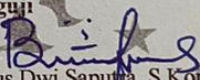
Lamongan, 14 Agustus 2023

Menyetujui,

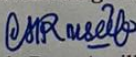
Pembimbing I

  
Muhammad Shodiq, S.Kom., M.Kom  
NIDN. 0721099401


Penguji

  
Bagus Dwi Saputra, S.Kom., M.Kom  
NIDN. 0724099402


Pembimbing II

  
Mala Rosa Aprillya, S.Kom., M.Kom  
NIDN. 0701049502

Ketua Program Studi  
Teknik Komputer

  
Mufti Ari Bianto, S.Kom., M.Kom  
NIDN. 0710069501

Mengetahui,  
Dekan Fakultas Sains  
Teknologi dan Pendidikan

  
Eko Handoyo, S.Kom., M.Kom  
NIDN. 0717029104

## HALAMAN PENGESAHAN

Skripsi dengan judul :

### **SISTEM MONITORING KADAR GAS KARBON DIOKSIDA (CO<sub>2</sub>) BERBASIS INTERNET OF THINGS**

Oleh:

**M. ARGHA WIRAYUDHA**

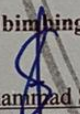
1903010021

Telah dilakukan pembimbingan Skripsi dan dinyatakan layak untuk mengikuti ujian skripsi pada Program Studi Teknik Komputer Fakultas Sains, Teknologi, dan Pendidikan Universitas Muhammadiyah Lamongan

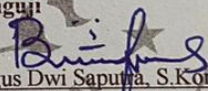
Lamongan, 14 Agustus 2023

Menyetujui,

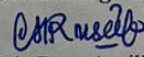
Pembimbing I

  
Muhammad Shodiq, S.Kom., M.Kom  
NIDN. 0721099401


Penguji

  
Bagus Dwi Saputra, S.Kom., M.Kom  
NIDN. 0724099402

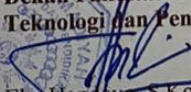
Pembimbing II

  
Mala Rosa Aprillya, S.Kom., M.Kom  
NIDN. 0701049502

**Ketua Program Studi  
Teknik Komputer**

  
Mufti Ari Bianto, S.Kom., M.Kom  
NIDN. 0710069501

Mengetahui,  
Dekan Fakultas Sains  
Teknologi dan Pendidikan

  
Eko Handoyo, S.Kom., M.Kom  
NIDN. 0717029104

## PERNYATAAN

Dengan ini saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar akademik di suatu perguruan tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Lamongan, 14 Agustus 2023



M. Agud Wirayudha

## **KATA PENGANTAR**

Alhamdulillah puji syukur kepada Allah SWT. Yang telah memberikan kemudahan dalam menyelesaikan skripsi ini. Sholawat dan salam dilimpahkan kepada junjungan kita Nabi Muhammad SAW dan semoga kita semua menerima syafaatnya di hari akhir, Aamiin. Dalam setiap langkah saya berusaha semaksimal mungkin untuk mewujudkan harapan-harapan yang saya impikan sebagai ungkapan terima kasih, saya ingin mempersembahkan skripsi ini untuk :

1. Bapak Eko Handoyo, S.Kom, M.Kom selaku Dekan Fakultas Sains, Teknologi dan Pendidikan Universitas Muhammadiyah Lamongan.
2. Bapak Mufti Ari Bianto, S.Kom., M.Kom selaku Ketua Program Studi Teknik Komputer Universitas Muhammadiyah Lamongan.
3. Muhammad Shodiq, S.Kom., M.Kom selaku Dosen Pembimbing I Skripsi yang telah memberikan pengarahan kepada penulis dalam penulisan.
4. Mala Rosa Aprillya, S.Kom., M.Kom selaku Dosen Pembimbing II Skripsi yang telah memberikan pengarahan kepada penulis dalam penulisan.
5. Orang Tua yang selalu memberikan dukungan materi, moral, dan doa yang tak putus untuk penulis. Terima kasih untuk semua pengorbanan, perhatian, dan kasih sayang yang diberikan selama ini.
6. Seluruh pihak yang telah membantu dalam pelaksanaan dan penyusunan Skripsi yang tidak dapat disebutkan satu persatu

Peneliti menyadari sepenuhnya atas keterbatasan yang ada, kemampuan, pengetahuan, waktu, pengalaman, maupun literatur yang tersedia maka tentunya semua ini jauh dari sempurna, oleh karena itu penulis mengharapkan kritik dan saran yang bersifat membangun dari pembaca sebagai perbaikan dimasa yang akan datang.

## DAFTAR ISI

|  | Halaman                             |
|--|-------------------------------------|
| HALAMAN JUDUL.....   | i                                   |
| HALAMAN PENGESAHAN.....                                      | ii                                  |
| HALAMAN PERNYATAAN PUBLIKASI .....                           | <b>Error! Bookmark not defined.</b> |
| HALAMAN PERNYATAAN .....                                     | iii                                 |
| KATA PENGANTAR .....   | iv                                  |
| DAFTAR ISI.....  | vi                                  |
| DAFTAR GAMBAR .....  | viii                                |
| DAFTAR TABEL.....  | ix                                  |
| DAFTAR LAMPIRAN .....  | x                                   |
| ABSTRAK .....  | xi                                  |
| <i>ABSTRACT</i> .....  | xii                                 |
| BAB I PENDAHULUAN .....                                      | 1                                   |
| 1.1 Latar Belakang .....                                     | 1                                   |
| 1.2 Rumusan Masalah .....                                    | 2                                   |
| 1.3 Batasan Masalah.....                                     | 2                                   |
| 1.4 Tujuan Penelitian.....                                   | 3                                   |
| 1.5 Manfaat Penelitian.....                                  | 3                                   |
| BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....                 | 4                                   |
| 2.1 Tinjauan Pustaka .....                                   | 4                                   |
| 2.2 Dasar Teori.....   | 7                                   |
| 2.2.1 <i>Internet of Things</i> (IoT).....                   | 7                                   |
| 2.2.2 Karbon Dioksida (CO <sub>2</sub> ) .....               | 8                                   |
| 2.2.3 <i>Sistem Monitoring</i> .....                         | 9                                   |
| 2.2.4 <i>Wireless Local Area Network</i> (WLAN).....         | 10                                  |
| 2.2.5 Sensor MQ-2 .....                                      | 11                                  |
| 2.2.6 NodeMCU ESP8266 .....                                  | 12                                  |
| 2.2.7 <i>Metode Waterfall</i> .....                          | 13                                  |
| 2.2.8 <i>Rule Based System</i> (RBS) .....                   | 15                                  |
| 2.2.9 <i>Object-Oriented Analysis and Design</i> (OOAD)..... | 16                                  |
| 2.2.10 <i>Use Case Diagram</i> .....                         | 17                                  |
| BAB III METODOLOGI PENELITIAN.....                           | 20                                  |
| 3.1 Bahan dan Alat Penelitian .....                          | 20                                  |
| 3.2 Prosedur Penelitian.....                                 | 20                                  |
| 3.2.1 Identifikasi Masalah .....                             | 21                                  |
| 3.2.2 Desain Sistem.....                                     | 21                                  |

|  |           |
|--|-----------|
| 3.2.3 Implementasi .....                           | 22        |
| 3.2.4 Pengujian.....                               | 22        |
| <b>BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....</b> | <b>23</b> |
| 4.1 Hasil Penelitian .....                         | 23        |
| 4.2.1 Identifikasi Masalah .....                   | 23        |
| 4.2.2 Desain Sistem .....                          | 24        |
| 4.2.3 Implementasi .....                           | 34        |
| 4.2. Pengujian.....                                | 41        |
| 4.2.1 Analisa Sistem.....                          | 41        |
| <b>BAB V PENUTUP.....</b>                          | <b>49</b> |
| 5.1 Kesimpulan.....                                | 49        |
| 5.2 Saran.....                                     | 49        |
| <b>DAFTAR PUSTAKA .....</b>                        | <b>51</b> |
| <b>LAMPIRAN .....</b>                              | <b>54</b> |

## DAFTAR GAMBAR

|  | Halaman |
|--|---------|
| Gambar 2.1 Sensor MQ-2 .....                                       | 12      |
| Gambar 2.2 Modul Sensor MQ-2.....                                  | 12      |
| Gambar 2.3 NodeMCU ESP8266 .....                                   | 13      |
| Gambar 2.4 Metode <i>Waterfall</i> .....                           | 14      |
| Gambar 3.1 Prosedur Penelitian.....                                | 21      |
| Gambar 4.1 <i>Use Case Diagram</i> .....                           | 25      |
| Gambar 4.2 Blok Diagram .....                                      | 26      |
| Gambar 4.3 <i>Flowchart Sistem</i> .....                           | 27      |
| Gambar 4.4 <i>Flowchart Rule Base Sensor</i> .....                 | 28      |
| Gambar 4.5 Denah Alat.....   | 30      |
| Gambar 4.6 Rancangan Alat .....                                    | 31      |
| Gambar 4.7 Rancangan Antarmuka .....                               | 32      |
| Gambar 4.8 Sensor MQ-2 dan Mikrokontroler ESP8266 yang Mati .....  | 35      |
| Gambar 4.9 Sensor MQ-2 dan Mikrokontroler ESP8266 yang Hidup ..... | 36      |
| Gambar 4.10 Implementasi Antarmuka .....                           | 37      |
| Gambar 4.11 Implementasi Antarmuka .....                           | 38      |
| Gambar 4.12 Kode Implementasi Antarmuka .....                      | 39      |
| Gambar 4.13 Kode Implementasi Antarmuka .....                      | 39      |



## DAFTAR TABEL

|  | Halaman |
|--|---------|
| Tabel 2.1 Kriteria Kualitas Udara (Amsar dkk, 2020).....                   | 8       |
| Tabel 2.2 Simbol <i>Use Case Diagram</i> (Febriani, 2023) .....            | 19      |
| Tabel 4. 1 Komponen Perangkat Keras.....                                   | 34      |
| Tabel 4.2 Jarak sensor MQ-2 Sejauh 10 cm dari Sumber Gas Karbon Dioksida . | 41      |
| Tabel 4.3 Jumlah data yang diambil dari pengujian di Gedung D.....         | 48      |

## DAFTAR LAMPIRAN

|   | Halaman |
|---|---------|
| Lampiran 1. <i>Code System (Sketch Arduino IDE)</i> ..... | 54      |
| Lampiran 2. <i>Code System (HTML Arduino IDE)</i> .....   | 57      |

**SISTEM *MONITORING* KADAR GAS KARBON DIOKSIDA (CO<sub>2</sub>)  
BERBASIS *INTERNET OF THINGS***

**ABSTRAK**

Dalam penelitian ini, dibuatlah sebuah sistem *monitoring* dengan konsep mengusung tentang pemantauan dan pengawasan terhadap eksistensi kadar gas karbon dioksida pada Gedung D yang bertempat di Universitas Muhammadiyah Lamongan, peneliti mengungkap kondisi daripada Gedung D tersebut, yaitu banyaknya aktivitas yang dilakukan pada tempat tersebut, sehingga dipastikan kualitas udara pada Gedung D juga memiliki kadar yang berbeda daripada tempat lainnya, terutama eksistensi dari karbon dioksida yang merupakan gas yang cukup berbahaya jika terpapar dalam rentan waktu tertentu. Berdasarkan aspek tersebut ditakutkan kadar gas karbon dioksida dapat mengancam kesehatan masyarakat luas pada kampus tersebut. Dengan adanya sistem yang telah berhasil dibuat dan berfokus pada kesehatan masyarakat ini, peneliti menggunakan sensor MQ-2 sebagai alat untuk pendeteksi gas karbon dioksida dengan menggunakan Mikrokontroler ESP8266 sebagai alat pemroses data yang di terima oleh sensor dengan modul Wi-Fi nya sebagai media pengirim data. Kemudian, data pada mikrokontroler tersebut akan dikirim menuju *Arduino IDE* untuk diproses dengan menggunakan *rule based* sebagai metode untuk menentukan ambang batas pada gas karbon dioksida tersebut yang kemudian akan dikirim kembali pada Mikrokontroler ESP8266 yang hasil akhirnya akan ditampilkan pada *website* sederhana dengan hasil kadar dan grafik yang *realtime* dan *buzzer* akan bunyi berdasarkan batasan tertinggi dari kadar karbon dioksida tersebut. Kesimpulan dari penjelasan tersebut adalah sistem *monitoring* ini telah dapat digunakan sebagaimana mestinya dan dapat dipertanggungjawabkan oleh masyarakat kampus Universitas Muhammadiyah Lamongan.

**Kata Kunci** : Mikrokontroler ESP8266, MQ-2, *Arduino IDE*, Karbon Dioksida.

**MONITORING SYSTEM FOR CARBON DIOXIDE (CO<sub>2</sub>) GAS LEVEL  
BASED ON THE INTERNET OF THINGS**

**ABSTRACT**

*In this study, a monitoring system was created with the concept of carrying out monitoring and supervision of the existence of carbon dioxide gas levels in Building D which is located at Muhammadiyah University of Lamongan, researchers revealed the condition of Building D, namely the many activities carried out in that place, so it is certain that the air quality in Building D also has different levels than other places, especially the existence of carbon dioxide which is a gas that is quite dangerous if exposed in a certain time. Based on this aspect, it is feared that carbon dioxide gas levels can threaten the health of the wider community on the campus. With a system that has been successfully created and focuses on public health, researchers use the MQ-2 sensor as a tool for detecting carbon dioxide gas using the ESP8266 Microcontroller as a data processing tool received by the sensor with its Wi-Fi module as a data sending medium. Then, the data on the microcontroller will be sent to the Arduino IDE to be processed using rule-based as a method for determining the threshold for carbon dioxide gas which will then be sent back to the ESP8266 Microcontroller which the final results will be displayed on a simple website with the results of realtime levels and graphs and a buzzer will sound based on the highest limit of carbon dioxide levels. The conclusion from this explanation is that this monitoring system can be used properly and can be accounted for by the campus community of Universitas Muhammadiyah Lamongan.*

**Keywords :** *ESP8266 Microcontroller, MQ-2, Arduino IDE, Carbon Dioxide.*

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Gas Karbon Dioksida (CO<sub>2</sub>) memberikan kontribusi terbesar terhadap pemanasan global diikuti oleh gas *methane* (CH<sub>4</sub>). Lebih dari 75% komposisi gas emisi rumah kaca pada atmosfer merupakan CO<sub>2</sub>, sehingga apabila kontribusi CO<sub>2</sub> dari berbagai kegiatan dapat dikurangi secara signifikan maka ada peluang bahwa dampak pemanasan global terhadap perubahan iklim akan berkurang (Sucipto dkk., 2023). Akibat proses pembakaran yang meningkatnya karbon dioksida yang cukup tinggi dikhawatirkan dapat memicu terjadinya pemanasan global yang mengakibatkan terjadinya pencairan es pada kutub dan mengganggu keseimbangan ekosistem (Amsar dkk, 2020).

Universitas Muhammadiyah Lamongan berada pada tempat yang strategis, tepatnya di pusat Kabupaten Lamongan, dan memiliki akses yang mudah dijangkau. Namun, di sekitar area kampus terdapat banyak sekali pabrik dan juga lalu lalang kendaraan bermotor, dan hal tersebut dikhawatirkan dapat mengganggu kesehatan masyarakat, sehingga menyebabkan kandungan udara di sekitar area kampus cenderung lebih buruk. Oleh karena itu, perlu adanya sistem yang bisa memberikan informasi mengenai kualitas udara yang baik dan kualitas udara yang buruk.

Sistem *monitoring* merupakan langkah untuk mengkaji kegiatan yang dilaksanakan telah sesuai dengan rencana, mengidentifikasi masalah yang timbul agar langsung dapat diatasi, melakukan penilaian pola kerja dan manajemen yang digunakan sudah tepat untuk mencapai tujuan, mengetahui kaitan antara kegiatan dengan tujuan untuk memperoleh ukuran kemajuan (Ayu dkk, 2020).

*Internet of Things* (IoT) merupakan suatu kerangka kerja global untuk menghubungkan infrastruktur jaringan, yang menghubungkan objek fisik maupun virtual melalui eksploitasi data *capture* dan teknologi komunikasi. Kerangka kerja IoT ini mencakup jaringan yang sudah ada dan juga *internet* beserta pengembangannya. Dalam kerangka ini, tersedia kemampuan identifikasi objek dan

sensor, serta koneksi yang menjadi dasar bagi pengembangan layanan dan aplikasi kolaboratif yang berdiri secara independen. Selain itu, ditandai juga dengan tingkat otonomi yang tinggi dalam pengumpulan data, transfer acara, konektivitas dalam jaringan, dan interoperabilitas (Setiadi dkk, 2018).

Berdasarkan permasalahan tersebut perlu dikembangkan Sistem *Monitoring* Kadar Gas Karbon Dioksida CO<sub>2</sub> Berbasis *Internet of Things*, sistem ini memiliki penyimpanan *log* data, dan memberi notifikasi peringatan melalui tampilan *web*, sehingga dapat digunakan sebagai peringatan terhadap masyarakat akan bahaya daripada gas Karbon Dioksida CO<sub>2</sub>, dan diharapkan dapat meminimalisir terjadinya gangguan kesehatan secara merata.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, maka diperoleh rumusan masalah penelitian sebagai berikut:

1. Bagaimana merancang dan membangun sistem *monitoring* gas CO<sub>2</sub>?
2. Bagaimana memberikan informasi keadaan gas CO<sub>2</sub> pada Gedung D di Universitas Muhammadiyah Lamongan?

## 1.3 Batasan Masalah

Dalam penelitian ini, agar tidak menyimpang dari tujuan yang ingin dicapai maka pembahasan akan dibatasi sebagai berikut:

1. Penelitian ini hanya menggunakan metode *rule based*.
2. Sistem yang dibangun ini khusus untuk *monitoring* keadaan pada Gedung D di Universitas Muhammadiyah Lamongan.
3. Penelitian ini hanya membahas tentang sistem *monitoring* berbasis *website*.
4. Penelitian ini hanya menggunakan 1 buah sensor MQ-2 untuk mendeteksi kadar gas CO<sub>2</sub> dengan radius dari sensor sekitar 10 cm dan pengambilan data setiap 1 detik sekali.
5. Penelitian ini hanya menggunakan *buzzer* sebagai perangkat keras berupa notifikasi suara.

#### **1.4 Tujuan Penelitian**

Tujuan dibangunnya penelitian sistem *monitoring* ini adalah:

1. Menghasilkan sistem deteksi dan *monitoring* kadar gas CO<sub>2</sub> pada Gedung D di Universitas Muhammadiyah Lamongan.
2. Menghasilkan informasi kadar bahaya bagi kesehatan melalui *website* sederhana dan memberikan informasi perubahan gas karbon dioksida dengan tampilan grafik dan *realtime* setiap 1 detik sekali sesuai perubahan waktu nyata.

#### **1.5 Manfaat Penelitian**

Manfaat dari penelitian yang dilakukan adalah sebagai berikut:

1. Membantu memantau kadar gas CO<sub>2</sub>.
2. Memberikan informasi kadar gas CO<sub>2</sub> secara *realtime*.
3. Dapat dijadikan referensi untuk penulisan selanjutnya.

## **BAB II**

### **TINJAUAN PUSTAKA DAN DASAR TEORI**

#### **2.1 Tinjauan Pustaka**

Dalam penelitian yang berjudul Rancang Bangun Sistem Pendeteksi Kebakaran Berbasis *Internet of Things* (IoT) Dan SMS Gateway Menggunakan *Arduino* (Mahendra dan Sasmoko, 2017). Keadaan alam di daerah PTPN IX memang berada pada dataran tinggi, sehingga memiliki kontur permukaan tanah yang berbukit, beberapa kali terjadi kebakaran hutan yang tidak dapat ditanggulangi sebab kurangnya informasi. Oleh karena itu, diperlukan sistem yang mampu mengirim data yang diperoleh sensor dari jarak jauh. Metode pengiriman data dilakukan menggunakan metode IoT. Dengan memakai *Arduino* yang terhubung dengan sensor suhu, asap, api dan *temperature* yang dikoneksikan dengan *internet* melalui *Subscriber Identity Module* (SIM) 900. Pengiriman data melalui IoT ini mempercepat pengiriman data mengenai kondisi lingkungan yang terjadi kebakaran, sehingga informasi kebakaran dapat diketahui lebih cepat oleh pihak berwenang. Rancang bangun Sistem pendeteksi kebakaran berbasis IoT dan *Short Message Service* (SMS) gateway ini sangat membantu memberikan informasi yang cepat untuk mengetahui kebakaran yang terjadi di hutan, dengan menggunakan metode *Internet of Things*, maka petugas akan dapat mengetahui kondisi secara *realtime*. Dikarenakan, teknologi ini mampu *me-monitoring hardware* menggunakan sarana komunikasi *internet* sehingga jarak dan lokasi tidak terpengaruh, asalkan sensor yang dipakai mendeteksi perubahan yang terjadi masih berfungsi dengan baik. Kekurangan dari penelitian ini yaitu, perlengkapan akan mati total jika terlalap api. Oleh karena itu, peralatan ini memerlukan suatu lapisan yang tahan terhadap api, sehingga sensor dapat tahan api dan disarankan tidak memakai sensor api model *flame* sensor 1 titik. Sebab, sensor ini kurang mampu mendeteksi api dari jarak yang jauh, sehingga tidak efektif jika ada kebakaran yang besar, selain itu sensor api 1 titik ini kurang dapat mendeteksi sinar panas matahari yang terlalu panas dan juga intensitas api, sehingga tingkat akurasi sangat kurang.



Hasil penelitian selanjutnya yang berjudul Perancangan Sistem *Monitoring Ketinggian Air Sebagai Pendeteksi Banjir Berbasis IoT Menggunakan Sensor Ultrasonik* (Pratama dkk, 2020). Curah hujan yang tinggi pada musim penghujan dapat berlangsung sekitar enam bulan, hal ini membuat hujan terus menerus mengguyur wilayah tertentu dan menyebabkan meningkatnya *volume* air yang memungkinkan potensi terjadinya banjir, sedangkan masyarakat tidak mendapat informasi atau pemberitahuan secara langsung ketika hal tersebut akan terjadi. Untuk mengantisipasi permasalahan tersebut, perancangan sistem *monitoring* ketinggian air sebagai pendeteksi banjir secara dini berbasis IoT dinilai efisien dan relevan terhadap kondisi terkini. Sistem akan mendeteksi jarak sensor dengan permukaan air, jika jarak permukaan air mendekati sensor maka sistem akan membunyikan sirine sebagai pertanda akan terjadi banjir. Data yang telah disimpan di *database* kemudian akan ditampilkan pada halaman *website* secara *realtime*. Rata-rata waktu data yang dikirimkan sensor ke *database* adalah 5 detik. Perhitungan waktu dibantu dengan alat *stopwatch* (Pratama dkk, 2020). Berdasarkan penelitian dan pengujian yang telah dilakukan terdapat kekurangan dari penelitian ini, hasil kesimpulan menjabarkan bahwa, koneksi *internet* yang tinggi dan stabil dibutuhkan agar *NodeMCU* dapat terhubung dengan *server*. Dari 100 data jarak permukaan air dengan sensor ultrasonik yang telah di ujicoba, mendapat 6 kali kesalahan pada pembacaan jarak dengan nilai kesalahan 1 cm sebanyak 5 kali dan 19 cm sebanyak 1 kali. Hasil pengujian sistem dari beberapa data jarak yang di *input*, semua data dapat disimpan pada *database* dan ditampilkan pada halaman *website* secara *realtime*. Rata-rata waktu data yang dikirimkan sensor ke *database* adalah 5 detik.

Hasil dari penelitian yang berjudul Perancangan Alat Pendeteksi CO<sub>2</sub> Menggunakan Sensor MQ-2 Berbasis *Internet of Things* yaitu, salah satu gas yang sangat berbahaya bagi kesehatan manusia adalah gas CO<sub>2</sub>, untuk mengetahui kadar di tempat tertentu yang mengandung CO<sub>2</sub> tanpa menggunakan alat bantu pendeteksi akan sulit dilakukan, karena gas CO<sub>2</sub> tersebut merupakan zat yang tidak dapat dilihat langsung oleh mata manusia, maka dalam penelitian ini dirancang sebuah

alat pendeteksi CO<sub>2</sub> yang dapat memberikan notifikasi kadar CO<sub>2</sub> ke *Telegram* yang berbasis *Internet Of Things*. Dalam penelitian ini, alat yang digunakan untuk pengukuran kadar CO<sub>2</sub> yaitu dengan menggunakan sensor MQ-2 sebagai alat pendeteksi kadar karbon dioksida, sehingga kadar CO<sub>2</sub> yang terdeteksi oleh sensor selanjutnya akan dikirim melalui jaringan IoT berbasis notifikasi via *Telegram*, notifikasi pada *Telegram* sebagai media pemberi informasi pada alat yang dibangun ini dengan 5 (lima) kategori yaitu: baik, sedang, tidak sehat, sangat tidak sehat dan berbahaya (Amsar dkk, 2020). Kekurangan dari penelitian ini yaitu, proses pengolahan data dan pengiriman data yang sudah terdeteksi oleh sensor dikirim via *Telegram* yang dibuat sangat tergantung dengan layanan jaringan *internet*, proses kecepatan layanan juga bergantung pada kekuatan *signal* dari *hotspot* atau *Wireless Fidelity* (Wi-Fi).

Hasil dari penelitian yang berjudul *Internet of Things* (IoT) Pada Prototipe Pendeteksi Kebocoran Gas Berbasis MQ-2 Dan SIM800L. *Internet* merupakan sumber informasi dan komunikasi yang sangat populer dewasa ini. Komunikasi dengan benda dapat dilakukan melalui *internet* yang dikenal dengan nama *Internet of Things*. Dalam Penelitian ini, merancang sebuah prototipe pendeteksi kebocoran gas dengan bentuk dan ukuran sekecil mungkin, dengan menggunakan alat mikrokontroler *Arduino Nano*, sensor gas MQ-2 dan *Global System for Mobile Communication* (GSM), serta Modul SIM800L. Alat pendeteksi kebocoran gas LPG ini bekerja dengan cara mendeteksi gas yang bocor dan memberikan informasi melalui SMS. Alat ini diharapkan dapat digunakan sebagai alat pengaman yang cukup murah dan mudah dibuat serta digunakan oleh masyarakat (Sumardi dan Mulyati, 2018). Kekurangan dari penelitian yaitu, hanya menggunakan 1 sensor yaitu sensor gas jenis MQ-2. Pendeteksian hanya dilakukan pada 1 titik saja, dan jarak sumber gas dengan sensor pendeteksi tidak ditentukan.

Hasil dari penelitian yang berjudul *Sistem Monitoring Gas CO<sub>2</sub> Di Gunung Dieng, Jawa Tengah Berbasis IoT*. Salah satu kebutuhan dasar manusia adalah rasa aman terhadap bencana alam yang ada di lingkungan sekitar. Sebagai contoh bencana gunung api yang kapan saja bisa mengamuk dengan mengeluarkan letusan berupa lava pijar, gas CO<sub>2</sub>, belerang, awan panas atau yang biasa disebut wedus

gembel. Hal inilah yang kemudian oleh Badan Penyelidikan dan Pengembangan Teknologi Kebencanaan dan Geologi (BPPTKG) Yogyakarta untuk memperbaharui sistem tersebut dengan menyediakan sebuah sistem informasi berbasis IoT untuk *monitoring* gas CO<sub>2</sub> pada Gunung Dieng yang menggunakan beberapa sensor. Sistem informasi ini akan dipadukan dengan *web server* yang akan menerima semua data-data sensor dari perangkat mini *Personal Computer* (PC) yaitu *Onion Omega 2* (Khoiruddin, 2018). Kekurangan dari penelitian ini yaitu, data yang diambil oleh sensor menggunakan data *dummy* (bukan data CO<sub>2</sub> pada Gunung Dieng yang sebenarnya). Sistem yang dibangun khusus untuk *monitoring* keadaan dari kawah Timbang pada Gunung Dieng.

## 2.2 Dasar Teori

### 2.2.1 *Internet of Things* (IoT)

*Internet of Things* (IoT) merupakan suatu ide yang bertujuan untuk memanfaatkan konektivitas *internet* secara lebih luas, dengan tujuan agar berbagai objek dapat terus-menerus terhubung. Dalam konsep serupa, *Internet of Things* diartikan sebagai suatu gagasan di mana objek-objek memiliki kemampuan untuk mengirimkan dan menerima data melalui 6 (enam) jenis jaringan *internet* yang berbeda, tanpa memerlukan campur tangan dari interaksi manusia ke manusia atau manusia ke komputer. Dengan kata lain, IoT memungkinkan objek-objek untuk berkomunikasi dan bertukar informasi secara otomatis melalui *internet*, membuka peluang baru untuk efisiensi dan keterhubungan dalam berbagai aspek kehidupan sehari-hari (Khoiruddin, 2018).

Secara prinsip, ide dasar dari IoT mencakup penggunaan sensor sebagai perangkat utama dalam mengumpulkan data, sambungan dengan *internet* sebagai medianya, serta *server* sebagai tempat pengumpulan informasi yang diterima dari sensor. Informasi tersebut nantinya dapat dianalisis guna menghasilkan ramalan yang akurat dan bermanfaat (Saputra dan Shodiq, 2022). Konsep mendasar dari *Internet of Things* (IoT) pertama kali dikemukakan oleh Kevin Ashton pada tahun 1999. Konsep ini kemudian menjadi landasan bagi berbagai perusahaan awal, seperti *Intel*, *Microsoft*, *Oracle* dan masih banyak lagi, untuk menjalankan inisiatif

mereka dalam mengembangkan teknologi IoT. Presentasi-presentasi yang kuat dan inspiratif dari perusahaan-perusahaan tersebut memainkan peran penting dalam mendorong pertumbuhan dan adopsi IoT di seluruh dunia. Ide brilian Kevin Ashton mengenai IoT telah membuka jalan bagi transformasi digital yang mendalam, di mana benda-benda dapat saling berhubungan dan berkomunikasi melalui jaringan, menghasilkan dampak luar biasa dalam berbagai industri dan aspek kehidupan manusia (Efendi, 2018).

### 2.2.2 Karbon Dioksida (CO<sub>2</sub>)

Karbon dioksida atau yang biasa disebut dengan (CO<sub>2</sub>) merupakan sebuah gas yang tidak memiliki warna dan tidak pula beracun pada konsentrasi kecil. Gas karbon dioksida CO<sub>2</sub> ketika berada di dalam atmosfer yakni sekitar 0,03 % mol dan ketika di dalam napas, gas karbon dioksida dihasilkan dari oksidasi biologis dari substansi makanan. Karena pada dasarnya gas karbon dioksida ketika berada di udara luar yaitu sekitar 1,5 % mol lebih besar. Gas karbon dioksida CO<sub>2</sub> cenderung berkumpul di daerah wilayah yang rendah dan ketika dalam keadaan kurang udara, maka akan dapat menyebabkan aspiksiasi (oleh pengeluaran oksigen). Sifat dari pengeluaran gas CO<sub>2</sub> tersebut dapat digunakan dalam pemadaman api (Khoiruddin, 2018). Kriteria udara berdasarkan Indeks Standar Pencemaran Udara (ISPU) dalam satuan *Part Per Million* (ppm) dapat dilihat pada Tabel 2.1.

Tabel 2.1 Kriteria Kualitas Udara (Amsar dkk, 2020).

| No. | Kategori | Rentang ppm | Penjelasan  | Efek   |
|-----|----------|-------------|---|--|
| 1.  | Baik     | 0 - 150     | Tingkat kualitas udara yang tidak memberikan efek bagi kesehatan manusia atau hewan dan tidak berpengaruh pada tumbuhan, bangunan ataupun nilai estetika.       | Tidak ada efek.                              |
| 2.  | Sedang   | 151 - 250   | Tingkat kualitas udara yang tidak memberikan efek bagi kesehatan manusia atau hewan, tetapi berpengaruh pada tumbuhan yang <i>sensitive</i> dan nilai estetika. | Perubahan kimia darah tapi tidak terdeteksi. |

Tabel 2.1 Kriteria Kualitas Udara (lanjutan)

| No. | Kategori           | Rentang    | Penjelasan   | Efek   |
|-----|--------------------|------------|--|--|
| 3   | Tidak Sehat        | 251 - 350  | Tingkat kualitas udara yang bersifat merugikan pada manusia ataupun kelompok hewan yang sensitif atau bisa menimbulkan kerusakan pada tumbuhan ataupun nilai estetika. | Peningkatan pada <i>kardiovaskular</i> yang berpenyakit jantung.   |
| 4   | Sangat Tidak Sehat | 351 - 450  | Tingkat kualitas udara yang dapat merugikan kesehatan pada sejumlah segmen populasi yang terpapar.   | Meningkatnya <i>kardiovaskular</i> pada yang berpenyakit jantung, dan akan tampak beberapa kelemahan yang terlihat secara nyata. |
| 5   | Berbahaya          | $\geq 451$ | Tingkat kualitas udara berbahaya yang secara umum dapat merugikan kesehatan yang serius pada populasi.   | Tingkat yang berbahaya bagi semua populasi yang terpapar.  |

### 2.2.3 Sistem *Monitoring*

Sistem *monitoring* merupakan serangkaian kegiatan yang bertujuan untuk menghimpun informasi dari berbagai sumber daya yang berbeda. Fokus utamanya adalah untuk mendapatkan data secara *real-time* guna memonitor performa serta keadaan suatu sistem atau lingkungan tertentu. Proses *monitoring* ini terdiri dari beberapa langkah kunci yang dapat dikelompokkan menjadi tiga tahap inti, yakni akuisisi data *monitoring*, analisis data *monitoring*, dan visualisasi data *monitoring* (Khoiruddin, 2018).

Dalam pengumpulan data *monitoring*, informasi diambil dari berbagai sumber seperti perangkat keras (*hardware*) yang terlibat, arus lalu lintas jaringan (*network traffic*), dan elemen-elemen lain yang relevan. Langkah selanjutnya yaitu proses analisis data *monitoring* di mana data yang telah terkumpul dianalisis secara mendalam. Analisis ini bertujuan untuk mengidentifikasi tren, pola, atau anomali yang mungkin muncul dalam data. Seluruh rangkaian proses ini berjalan sebagai layanan (*service*) yang beroperasi secara *real-time* dalam interval waktu yang telah ditetapkan sebelumnya (Khoiruddin, 2018).

Proses-proses tersebut dimulai dengan mengumpulkan data mentah, yang kemudian mengalami tahap analisis mendalam untuk menggali informasi berharga. Dengan demikian, sistem *monitoring* bukan hanya tentang pengamatan pasif, tetapi juga melibatkan interpretasi dan pemahaman yang lebih dalam terhadap data yang terkumpul. Setelah melalui proses analisis, data yang telah diolah akan disajikan melalui berbagai tampilan grafis atau *dashboard* yang mudah dipahami. Hal ini memungkinkan pengguna sistem *monitoring* untuk dengan cepat mengenali perubahan atau situasi yang memerlukan perhatian khusus (Khoiruddin, 2018).

Dengan demikian, sistem *monitoring* bukan hanya tentang memantau, tetapi juga melibatkan pengumpulan data, analisis cerdas, dan visualisasi yang efektif untuk memberikan pemahaman yang lebih baik tentang status dan kinerja suatu sistem. Semua langkah ini dilakukan secara berkesinambungan guna menjaga operasional yang lancar dan mendeteksi potensi masalah sebelum berdampak lebih besar (Khoiruddin, 2018).

#### **2.2.4 Wireless Local Area Network (WLAN)**

Jaringan komputer nirkabel, yang dikenal juga sebagai *Wireless Local Area Network* (WLAN) atau populer dengan sebutan *Wireless Fidelity* (Wi-Fi) merupakan suatu bentuk jaringan area lokal yang menggunakan teknologi gelombang radio untuk melakukan pertukaran data. Teknologi WLAN telah menjadi pilihan utama bagi pengguna komputer karena memberikan berbagai keuntungan, kenyamanan, serta fleksibilitas yang luar biasa. Dengan adanya WLAN, pengguna memiliki kemampuan untuk mengakses jaringan komputer dan *internet* tanpa terikat oleh kabel yang sebelumnya diperlukan untuk terhubung (Asteroid dkk., 2016).

Daya tarik utama dari teknologi WLAN adalah kemampuan untuk bergerak secara bebas dalam jangkauan sinyal Wi-Fi tanpa keharusan mengandalkan konektivitas kabel. Hal ini memungkinkan pengguna untuk dengan mudah berpindah lokasi, baik dalam ruangan maupun antar-ruangan, tanpa kehilangan akses ke jaringan. Fleksibilitas ini menjadi aspek penting, terutama dalam lingkungan yang terus berubah dan seringkali memerlukan adaptasi cepat.

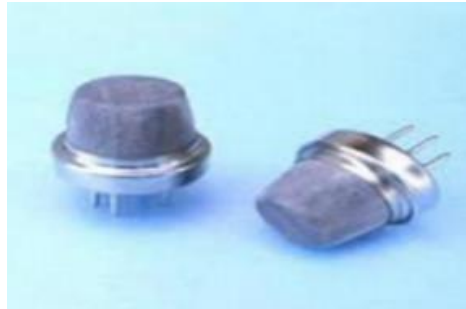
Kebebasan yang ditawarkan oleh WLAN juga memberikan kenyamanan bagi pengguna perangkat seluler seperti laptop, *smartphone*, dan tablet. Pengguna dapat tetap terhubung dengan *internet* saat berada di area Wi-Fi, tanpa harus tergantung pada jaringan seluler yang mungkin memiliki keterbatasan cakupan atau kuota data. Hal tersebut ini membuat aktivitas dalam menjelajahi *internet*, bekerja, atau bersosialisasi secara *online* menjadi lebih lancar dan mudah diakses (Asteroid dkk., 2016).

Selain itu, pengembangan teknologi WLAN juga berdampak positif pada mobilitas dan produktivitas. Dalam lingkungan bisnis, WLAN memungkinkan karyawan untuk bekerja dari berbagai lokasi di dalam gedung kantor tanpa gangguan konektivitas. Di tempat umum seperti bandara, kafe, dan pusat perbelanjaan, ketersediaan Wi-Fi gratis meningkatkan nilai tambah bagi pelanggan yang ingin tetap terhubung sambil menikmati layanan yang ditawarkan (Asteroid dkk., 2016).

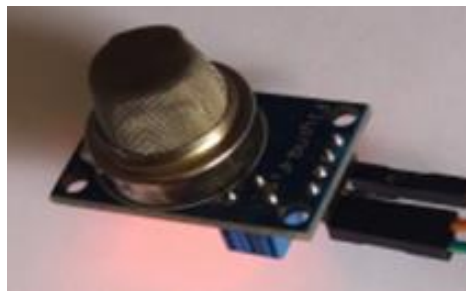
Dengan terus berkembangnya teknologi dan peningkatan kecepatan serta keamanan jaringan WLAN, diharapkan penggunaan Wi-Fi akan terus meluas, menciptakan ekosistem yang semakin terhubung dan mendukung berbagai aspek kehidupan *modern* (Asteroid dkk., 2016).

### **2.2.5 Sensor MQ-2**

Salah satu sensor yang sangat sensitif terhadap asap adalah sensor MQ-2, sensor ini mampu mendeteksi gas dari hasil pembakaran, sehingga asap di udara dari hasil pembakaran tersebut dapat dibawa sebagai tegangan analog, sensor MQ-2 ini juga mampu mendeteksi kebocoran gas. Apabila terdapat kebocoran gas, konduktivitas sensor akan berubah menjadi lebih tinggi, konduktivitas sensor ikut naik setiap terjadinya perubahan pada konsentrasi gas. MQ-2 sensitif terhadap gas LPG, *Propane* ( $C_3H_8$ ), *Hidrogen* (H), Karbon Monoksida (CO), *Methane* ( $CH_4$ ) dan *Alcohol* (-OH) serta gas mudah terbakar udara lainnya (Amsar dkk, 2020). Sensor MQ-2 dapat dilihat pada Gambar 2.1 dan 2.2.



Gambar 2.1 Sensor MQ-2



Gambar 2.2 Modul Sensor MQ-2

Sensor gas dan asap ini memiliki kemampuan untuk mendeteksi konsentrasi gas yang mudah terbakar di dalam udara, serta mampu mengukur jumlah asap. Sensor ini menghasilkan keluaran berupa data dalam bentuk tegangan analog. Rentang jarak untuk mengukur konsentrasi gas pada sensor tipe MQ-2 ini adalah antara 300 hingga 10.000 ppm. Keunggulan lainnya adalah sensor ini dapat beroperasi dalam rentang suhu yang cukup luas, yaitu dari  $-20$  hingga  $50^{\circ}\text{C}$ . Dalam hal konsumsi daya, sensor ini sangat efisien, hanya mengonsumsi kurang dari 150 mA saat diberi tegangan 5V. Dengan fitur-fitur unggulannya ini, sensor ini merupakan pilihan yang handal dalam mendeteksi gas yang mudah terbakar dan konsentrasi asap dalam berbagai kondisi lingkungan, sehingga penggunaan sensor tersebut diharapkan dapat mendeteksi suatu gas sesuai dengan kebutuhan (Ardiansyah, 2020) (Amsar dkk, 2020).

### 2.2.6 NodeMCU ESP8266

NodeMCU ESP8266 merupakan sebuah modul mikrokontroler yang dirancang dengan memasukkan komponen ESP8266 ke dalamnya. ESP8266 memiliki peran utama dalam menyediakan konektivitas jaringan WiFi antara modul mikrokontroler dan jaringan WiFi yang ada. Hal ini memungkinkan modul tersebut



untuk terhubung dengan *internet* melalui jaringan WiFi (Pangestu dkk, 2019). Salah satu fitur menarik dari NodeMCU adalah Bahasa Pemrograman Lua yang digunakan sebagai basisnya. Selain itu, NodeMCU juga dapat diprogram menggunakan *Arduino IDE*, yang umumnya lebih dikenal oleh banyak pengembang. Fleksibilitas ini memungkinkan pengguna untuk memilih bahasa pemrograman yang paling sesuai dengan kebutuhan dan preferensi pengguna (Pangestu dkk, 2019).

Kelebihan utama dalam memilih NodeMCU ESP8266 sebagai pilihan modul mikrokontroler adalah kemudahan dalam proses pengembangannya. Modul ini menyediakan pin I/O yang memadai, memungkinkan pengguna untuk menghubungkan berbagai komponen eksternal dengan mudah. Selain itu, keberadaan modul ESP8266 dalam NodeMCU juga memungkinkan akses mudah ke jaringan *internet* melalui koneksi WiFi. Ini berarti modul ini dapat digunakan untuk mengirim atau mengambil data secara efisien melalui jaringan WiFi, membuka peluang untuk aplikasi berbasis *Internet of Things* yang lebih luas (Pangestu dkk, 2019).

Dengan demikian, NodeMCU ESP8266 merupakan pilihan yang sangat berguna dan serbaguna untuk proyek-proyek yang memerlukan konektivitas WiFi dan kontrol mikrokontroler yang efisien (Pangestu dkk, 2019).

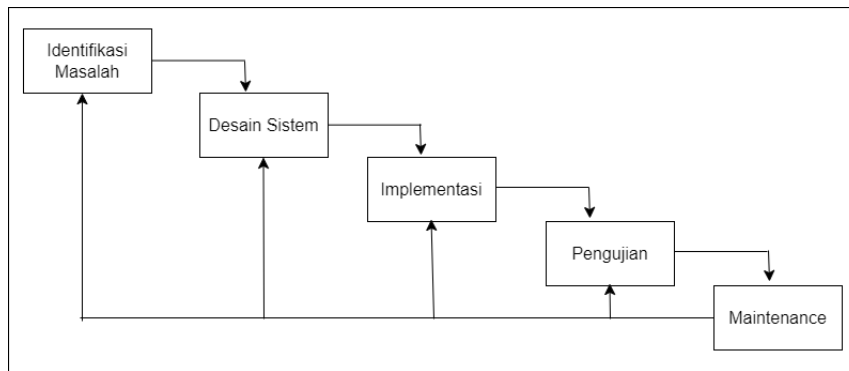


Gambar 2.3 NodeMCU ESP8266

### 2.2.7 Metode *Waterfall*

Metode *waterfall* merupakan model yang paling banyak digunakan untuk tahap pengembangan. Metode *waterfall* ini juga dikenal dengan model tradisional atau model klasik. Metode *waterfall* sering juga disebut model sekuensial linier

(*sequential linear*) atau alur hidup klasik (*classic cycle*). Metode *waterfall* ini menyediakan pendekatan alur hidup perangkat lunak secara sekuensial terurut dimulai dari analisis, desain, pengkodean, pengujian dan tahap pendukung (Susilo, 2018). Metode *waterfall* dapat dilihat pada Gambar 2.4.



Gambar 2.4 Metode *Waterfall*

Identifikasi masalah, metode pengumpulan informasi ini dapat diperoleh dengan berbagai macam cara diantaranya, diskusi, observasi, survei, wawancara, dan sebagainya. Informasi yang diperoleh kemudian diolah dan dianalisa sehingga didapatkan data atau informasi yang lengkap mengenai spesifikasi kebutuhan pengguna akan perangkat lunak yang akan dikembangkan (Susilo, 2018).

Desain sistem, informasi mengenai spesifikasi kebutuhan dari tahap *Requirement Analysis* selanjutnya di analisa pada tahap ini untuk kemudian diimplementasikan pada desain pengembangan. Perancangan desain dilakukan dengan tujuan membantu memberikan gambaran lengkap mengenai apa yang harus dikerjakan (Susilo, 2018).

Implementasi, pembuatan perangkat lunak dibagi menjadi modul-modul kecil yang nantinya akan digabungkan dalam tahap berikutnya. Disamping itu, pada fase ini juga dilakukan pengujian dan pemeriksaan terhadap fungsionalitas modul yang sudah dibuat, apakah sudah memenuhi kriteria yang diinginkan atau belum (Susilo, 2018).

Pengujian, setelah seluruh unit atau modul yang dikembangkan dan diuji di tahap implementasi selanjutnya diintegrasikan dalam sistem secara keseluruhan. Setelah proses integrasi selesai, selanjutnya dilakukan pemeriksaan dan pengujian

sistem secara keseluruhan untuk mengidentifikasi kemungkinan adanya kegagalan dan kesalahan sistem (Susilo, 2018).

*Maintenance*, perangkat lunak yang sudah jadi dioperasikan pengguna dan dilakukan pemeliharaan. Pemeliharaan memungkinkan pengembang untuk melakukan perbaikan atas kesalahan yang tidak terdeteksi pada tahap-tahap sebelumnya. Pemeliharaan meliputi perbaikan kesalahan, perbaikan implementasi unit sistem, dan peningkatan dan penyesuaian sistem sesuai dengan kebutuhan (Susilo, 2018).

### **2.2.8 Rule Based System (RBS)**

*Rule Based System* (RBS) atau yang biasa disebut sistem berbasis aturan merupakan varian khusus dari sistem pakar yang terdiri dari sejumlah aturan yang telah ditetapkan sebelumnya. Dalam implementasinya, sistem ini dapat dikembangkan dengan memanfaatkan pengetahuan mendalam dari para ahli di bidang tersebut atau melalui proses pembelajaran dari data empiris yang ada (Cahyanto dkk., 2020). Penerapan *rule based system* bertujuan untuk mengatasi segala masalah dan kendala baik secara teknis maupun secara non-teknis yang dapat berpengaruh dalam kinerja sistem berdasarkan pengumpulan data dan penguasaan data empiris telah terkumpul tersebut (Handoyo dkk, 2019) (Cahyanto dkk., 2020).

Dalam esensinya, *rule based system* merupakan suatu model yang mengambil keputusan atau memberikan solusi tergantung pada kumpulan aturan logika yang telah didefinisikan sebelumnya. Setiap aturan umumnya terdiri dari kondisi-kondisi tertentu yang mengarah ke tindakan atau hasil spesifik. Penggunaan sistem ini sangat berguna dalam *domain* di mana keputusan dapat diambil berdasarkan serangkaian instruksi yang jelas dan terstruktur (Cahyanto dkk., 2020). Sehingga, untuk melakukan sebuah klasifikasi kondisi membutuhkan metode yang tepat dalam mengelola pengetahuan yang diadopsi dari pakar sehingga diperoleh hasil yang akurat pada sistem tersebut (Bianto, 2019) (Cahyanto dkk., 2020).

*Rule Based System* dapat dikembangkan dengan dua pendekatan utama. Pertama, dengan memanfaatkan pengetahuan eksplisit yang dimiliki oleh para ahli di bidang tersebut. Aturan-aturan ini dibangun berdasarkan pengetahuan mendalam

yang diperoleh dari pengalaman dan kepiawaian para ahli. Kedua, dengan cara mempelajari pola-pola dari data nyata yang ada. Dalam hal ini, sistem menggunakan teknik pembelajaran mesin untuk mengidentifikasi aturan-aturan yang dapat digeneralisasi dari data yang diberikan (Cahyanto dkk., 2020). Kelebihan *rule based system* meliputi kejelasan dalam pengambilan keputusan, penjelasan yang dapat ditelusuri mengapa suatu keputusan dibuat, dan kemampuan untuk menghasilkan solusi konsisten berdasarkan aturan yang telah ditetapkan. Namun, batasan utamanya adalah ketidakmampuan untuk menghadapi situasi yang tidak sesuai dengan aturan yang ada, serta kesulitan dalam menangani konteks yang kompleks atau ambigu (Cahyanto dkk., 2020).

Dalam perkembangannya, sistem berbasis aturan sering digabungkan dengan teknik-teknik lain seperti pembelajaran mesin atau kecerdasan buatan untuk meningkatkan kinerja dan adaptabilitasnya (Cahyanto dkk., 2020). Kombinasi dari penggunaan metode ini memungkinkan sistem untuk tetap memanfaatkan kejelasan aturan dari pengalaman baru atau situasi yang tidak terduga agar prosedur penelitian ini lebih terstruktur dan berjalan sistematis sesuai dengan rencana awal dari sistem tersebut (Aprillya dan Chasanah, 2022) (Cahyanto dkk., 2020).

### **2.2.9 Object-Oriented Analysis and Design (OOAD)**

*Object-Oriented Analysis and Design (OOAD)* merupakan pendekatan baru dalam menganalisis serta merancang solusi untuk masalah dengan menggunakan model yang dibentuk berdasarkan konsep dunia nyata. Pendekatan ini memiliki dasar utama pada objek, yang merupakan gabungan antara struktur data dan perilaku yang ada dalam suatu entitas tunggal. Dalam konteks ini, berorientasi objek menggambarkan bagaimana pengguna mengorganisasikan perangkat lunak sebagai sekumpulan objek khusus yang memiliki tidak hanya struktur data, tetapi juga perilaku yang saling terkait.

Dengan menerapkan pendekatan OOAD, para profesional IT dapat melihat permasalahan dari sudut pandang yang lebih intuitif dan sesuai dengan dunia nyata. Pendekatan ini memungkinkan mereka untuk merepresentasikan entitas-entitas dalam masalah secara langsung ke dalam objek-objek dalam perangkat lunak yang

direncanakan. Selain itu, OOAD juga memfasilitasi pemodelan hubungan dan interaksi antara objek-objek ini, sehingga solusi yang dihasilkan lebih mencerminkan kompleksitas dan dinamika hubungan dalam dunia riil (Neyfa dan Tamara, 2016).

Melalui OOAD, proses pengembangan perangkat lunak menjadi lebih terstruktur dan terorganisir. Pengembang dapat lebih mudah memahami bagaimana setiap objek berperan dalam menjalankan fungsi tertentu dan berinteraksi dengan objek lainnya. Dengan kata lain, OOAD membantu mengurai kompleksitas suatu masalah dengan membaginya menjadi elemen-elemen yang lebih kecil, yaitu objek-objek, yang nantinya dapat diimplementasikan dalam kode program. Pentingnya pendekatan OOAD terletak pada kemampuannya untuk menciptakan perangkat lunak yang lebih modular, fleksibel, dan mudah diubah. Dengan menjadikan objek sebagai komponen dasar, perubahan atau penambahan fitur dapat dilakukan dengan lebih terkendali dan minim dampaknya terhadap bagian lain dari perangkat lunak (Neyfa dan Tamara, 2016).

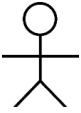
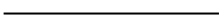
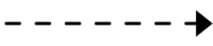
Oleh karena itu, OOAD tidak hanya sekadar tentang pemodelan, tetapi juga tentang bagaimana merancang struktur perangkat lunak agar sesuai dengan kebutuhan bisnis dan dapat beradaptasi dengan perubahan yang mungkin terjadi di masa depan. Secara keseluruhan, OOAD merupakan salah satu pendekatan yang kuat dalam mengatasi kompleksitas dari sebuah perangkat lunak dengan merujuk pada struktur dan interaksi objek dalam dunia nyata. Dapat disimpulkan bahwa hal tersebut bukanlah hanya tentang teknik pemrograman, tetapi juga tentang filosofi dalam merancang solusi yang tanggap, modular, dan berkelanjutan dalam menghadapi dinamika kebutuhan bisnis dan teknologi yang terus berkembang (Neyfa dan Tamara, 2016).

#### **2.2.10 Use Case Diagram**






*Use case diagram* merupakan sebuah penggambaran sebuah fungsi sistem dari sudut pandang pengguna eksternal dan dalam cara yang mudah dipahami. *Use case diagram* merupakan penyusunan kembali lingkup fungsional sistem yang disederhanakan lagi. *Use case diagram* merupakan diagram yang menggambarkan

interaksi antara sistem dengan sistem eksternal pengguna. *Use case diagram* merupakan titik awal yang baik dalam memahami dan menganalisis kebutuhan sistem pada saat perancangan. *Use case diagram* dapat digunakan untuk kebutuhan apa saja yang diperlukan dalam suatu sistem, sehingga sistem dapat digambarkan dengan visualisasi yang jelas bagaimana proses dari sistem tersebut, bagaimana cara aktor menggunakan sistem, serta apa saja yang dapat dilakukan pada suatu sistem yang akan dibuat (Steven dkk. 2020). *Use case diagram* menggambarkan interaksi yang saling terkait antara aktor, yang merupakan entitas yang berinteraksi dengan sistem, dan sistem itu sendiri. Tujuan utama dari aktor dalam *use case diagram* adalah untuk menjalankan berbagai aktivitas di dalam sistem yang akan dikelola. *Use case diagram* membantu dalam mengidentifikasi dan menggambarkan skenario-skenario yang melibatkan interaksi antara aktor dan sistem, sehingga mempermudah pemahaman tentang bagaimana sistem beroperasi dalam berbagai situasi yang berbeda. Dengan adanya *use case diagram*, tim pengembang dapat memiliki pandangan yang jelas tentang fungsionalitas yang diharapkan dari sistem dan bagaimana pengguna akan berinteraksi dengan sistem tersebut (Febriani, 2023). Simbol *use case diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol *Use Case Diagram* (Febriani, 2023)

| No. | Simbol  | Nama               | Keterangan   |
|-----|---|--------------------|--|
| 1.  |  | <i>Actor</i>       | Menspesifikasikan peran pengguna mainkan ketika berinteraksi dengan <i>usecase</i> .   |
| 2.  |  | <i>Association</i> | Apa yang menghubungkan antara objek satu dengan objek lain.  |
| 3.  |  | <i>Dependency</i>  | Hubungan dimana perubahan terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri. |

Tabel 2.2 Simbol *Use Case Diagram* (lanjutan)

| No. | Simbol  | Nama                  | Keterangan   |
|-----|---|-----------------------|--|
| 4   |    | <i>Generalization</i> | Hubungan dimana objek ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ). |
| 5   |    | <i>Include</i>        | Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.   |
| 6   |    | <i>Extend</i>         | Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.                |
| 7   |   | <i>System</i>         | Menspesifikasikan paket yang menampilkan sistem secara terbatas.   |
| 8   |  | <i>Use Case</i>       | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan  |

## **BAB III**

### **METODOLOGI PENELITIAN**

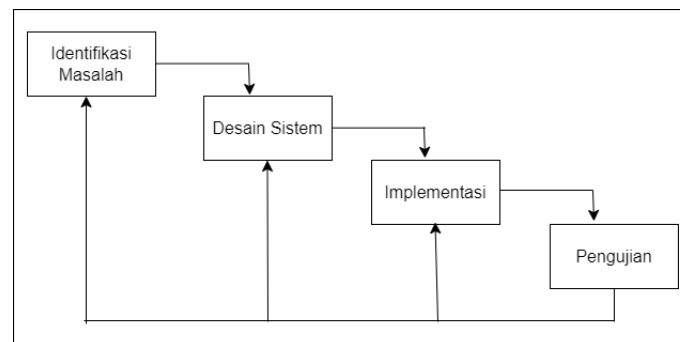
#### **3.1 Bahan dan Alat Penelitian**

Bahan penelitian ini menggunakan gas CO<sub>2</sub>. Kemudian, alat penelitian ini menggunakan *hardware* dan *software*. Untuk alat *hardware* berupa 1 sensor MQ-2 sebagai alat pendeteksi gas CO<sub>2</sub>, ESP8266 sebagai perangkat tambahan mikrokontroler seperti *Arduino IDE* agar dapat terhubung langsung dengan Wi-Fi dan membuat koneksi TCP/IP, ditambah adanya *Buzzer* yang merupakan perangkat keras yang berfungsi untuk memberi notifikasi berupa suara. Seperangkat komputer dengan spesifikasi *Intel(R) Core (TM) i3-6006U CPU @2.00GHz 1.99 GHz RAM 4,00 GB DDR4* dengan OS *Windows 10 pro 64-bit*, 6 kabel *jumper* sebagai koneksi *female-to-female* pada kedua ujung kabel. Untuk *software* menggunakan alat penelitian *Arduino IDE* sebagai *code system*, Bahasa C++ digunakan sebagai bahasa pemrograman, HTML digunakan untuk membuat kerangka dasar halaman *web*, termasuk bagaimana konten diorganisasikan secara hierarkis. Ini mencakup pembuatan judul, paragraf, daftar, tabel, dan lebih banyak elemen lainnya.

#### **3.2 Prosedur Penelitian**

Prosedur penelitian dilakukan dengan menggunakan metode *waterfall*. Metode *waterfall* adalah metode perangkat lunak dengan 5 (lima) langkah pemodelan. Lima fase tersebut meliputi analisis dan definisi kebutuhan, desain sistem, implementasi dan pengujian, integrasi dan pengujian sistem, serta pengoperasian dan pemeliharaan. Penerapan pengembangan model *waterfall* pada sistem *monitoring* kadar gas CO<sub>2</sub>, seperti yang ditunjukkan pada Gambar 3.1.





Gambar 3.1 Prosedur Penelitian

### 3.2.1 Identifikasi Masalah

Pada tahapan identifikasi masalah, penulis mengamati kondisi yang terdapat pada Gedung D di Universitas Muhammadiyah Lamongan dengan melihat kondisi kualitas udara, terutama gas karbon dioksida sebagai dasar dari pembuatan sistem *monitoring* ini. Berdasarkan hal itu, peneliti melakukan analisis yang akan menghasilkan sebuah solusi berupa pembuatan sistem *monitoring* gas CO<sub>2</sub> dengan menggunakan sensor MQ-2 dengan hasil akhir berupa *website* sederhana dibantu oleh *buzzer* sebagai alat notifikasi berupa suara, agar dosen dan mahasiswa serta para perangkat institusi dapat memantau dan memahami kondisi perubahan gas CO<sub>2</sub> di sekitar lingkungan Gedung D yang bertempat di Universitas Muhammadiyah Lamongan.

### 3.2.2 Desain Sistem

Pada langkah ini alat dibuat berdasarkan diagram desain yang telah dirancang sebelumnya. Proses ini meliputi penambahan komponen tambahan, perancangan sensor, dan pengembangan program khusus untuk perangkat. Setelah fase pembuatan alat selesai, alat tersebut akan diintegrasikan dengan *platform Arduino IDE* untuk membuat sistem yang berfungsi penuh dengan tujuan *monitoring* terhadap kadar gas karbon dioksida pada Gedung D di Universitas Muhammadiyah Lamongan. Selain itu, hasil akhir alat ini akan berupa sebuah tampilan pada situs *web* sederhana, agar pengguna dapat dengan mudah mengakses dan mengawasi kondisi gas karbon dioksida pada Gedung D di Universitas Muhammadiyah Lamongan secara *realtime*.

### 3.2.3 Implementasi

Pada langkah ini, implementasi sensor MQ-2 dilakukan dalam Bahasa Pemrograman C++ dan HTML pada *platform Arduino IDE* sebagai tempat untuk mengolah data dari *input* sensor karbon dioksida yaitu MQ-2. Setelah data berhasil diproses, maka akan ditampilkan pada sistem berbasis *web* sederhana yang telah terhubung dengan Mikrokontroler ESP8266 dibantu dengan perangkat keras tambahan berupa *buzzer*. Proses ini memungkinkan pengguna dalam pemantauan terhadap konsentrasi gas karbon dioksida secara *realtime* melalui antarmuka *web* yang responsif dan mudah diakses.

### 3.2.4 Pengujian

Pengujian sistem merupakan langkah akhir yang dilakukan peneliti dalam memastikan bahwa alat pemantau gas karbon dioksida dan sistem yang menggunakan sensor MQ-2 bekerja sesuai dengan rencana dan dapat berfungsi dengan baik dan relevan. Selain itu, pengujian ini bertujuan untuk mengidentifikasi dan menemukan kemungkinan kekurangan (*weakness*) atau kesalahan (*error*) pada alat dan sistem yang dikembangkan.

## BAB IV

### HASIL PENELITIAN DAN PEMBAHASAN

#### 4.1 Hasil Penelitian

##### 4.2.1 Identifikasi Masalah

Pada awal penelitian ini, peneliti wajib melakukan identifikasi masalah sebagai tahap awal penelitian. Peneliti perlu menjelaskan secara rinci identifikasi masalah yang akan diangkat dalam latar belakang masalahnya. Terdapat 2 (dua) metode yang dapat digunakan untuk identifikasi masalah, yaitu observasi dan studi literatur.

##### a) Observasi

Dalam pelaksanaan pengumpulan data untuk melengkapi penelitian ini, peneliti menggunakan teknik yang umum dalam kegiatan ilmiah, yaitu observasi. Observasi yang dilakukan oleh peneliti menghasilkan data berupa luas Gedung D dengan luas kurang lebih 110 m<sup>2</sup>, tinggi dari Gedung D yaitu kurang lebih 8 meter<sup>2</sup>, Gedung D terdiri dari 3 lantai, peneliti berencana melakukan *monitoring* gas karbon dioksida pada Gedung D di Universitas Muhammadiyah Lamongan.

Berdasarkan hasil observasi yang dilakukan, dapat disimpulkan bahwa sistem *monitoring* gas karbon dioksida menggunakan sensor MQ-2 memang relevan untuk ditempatkan pada Gedung D di Universitas Muhammadiyah Lamongan, peneliti melihat kondisi pada Gedung D tersebut yang cenderung terdapat banyak sekali aktivitas yang dilakukan banyak orang pada satu tempat terbuka, sehingga diharapkan sistem *monitoring* gas karbon dioksida ini dapat berjalan sesuai rencana dan relevan dengan kondisi pada tempat penelitian.

##### b) Studi Literatur

Kemudian, data berhasil dihimpun untuk sistem *monitoring* gas karbon dioksida (CO<sub>2</sub>) yang berbasis *Internet of Things* (IoT). Pengumpulan data melibatkan penerapan sensor gas karbon dioksida berupa MQ-2 yang terintegrasi dengan Mikrokontroler ESP8266. Tujuan dari pengumpulan data ini adalah untuk memperoleh pemahaman yang lebih komprehensif mengenai operasi sistem ini, serta hasil yang tercapai setelah berhasil membangun sistem *monitoring* gas karbon

dioksida berbasis IoT menggunakan sensor gas *ammonia* MQ-2 dapat digunakan sebagai acuan oleh peneliti untuk pengembangan sistem pada penelitian selanjutnya mengenai penelitian ini.

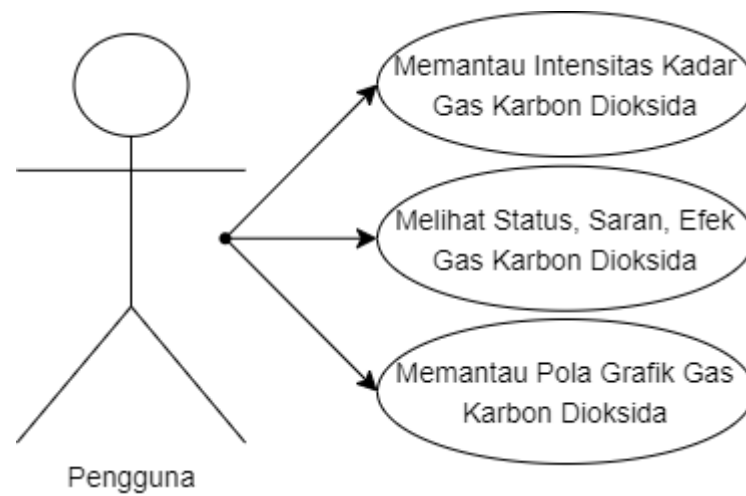
Sistem *monitoring* gas karbon dioksida ini dirancang untuk mengawasi kondisi gas karbon dioksida secara *realtime* dengan menggunakan koneksi *internet* sebagai media untuk mengakses *website* ini. Pengguna dapat mengawasi kondisi gas karbon dioksida dari jarak jauh melalui sebuah *website* sederhana yang telah dibuat untuk pengawasan terhadap perubahan gas karbon dioksida secara *realtime*, sehingga pengoperasian sistem diharapkan dapat dilaksanakan sesuai dengan kebutuhan pada penelitian ini.

#### **4.2.2 Desain Sistem**

Langkah selanjutnya merupakan desain sistem, yang berupa tahapan penting dalam pengembangan sistem ini. Desain sistem dari penelitian ini memiliki 4 (empat) tahapan, empat tahapan tersebut antara lain yaitu *use case diagram*, denah alat, rancangan alat, dan rancangan antarmuka dari sistem yang akan dibangun berdasarkan hasil pengumpulan data dan analisis sebelumnya.

##### a) *Use Case Diagram*

Langkah awal melibatkan pembuatan *use case diagram*, sebuah visualisasi grafis yang menggambarkan interaksi antara aktor atau pengguna dengan sistem. Tujuan dari diagram ini adalah untuk mengilustrasikan fungsionalitas sistem dari perspektif pengguna, membantu dalam menganalisis kebutuhan perangkat lunak, serta memperoleh pemahaman yang lebih baik. Dalam *use case diagram*, aktor merupakan entitas eksternal yang berhubungan dengan sistem, sementara *use case* atau kasus penggunaan merujuk pada tindakan atau fungsi yang dapat dijalankan oleh aktor di dalam sistem. Diagram ini mencerminkan hubungan dan interaksi antara aktor dan *use case* dalam berbagai skenario. *Use case diagram* juga membantu dalam memvisualisasikan cara pengguna dapat melaksanakan fungsi-fungsi pada sistem pemantauan gas karbon dioksida berbasis *Internet of Things* (IoT), seperti yang ditunjukkan pada Gambar 4.1.



Gambar 4.1 Use Case Diagram

Berdasarkan Gambar 4.1, pengguna dapat mengakses *website* dengan fitur sederhana yaitu tabel *monitoring* kadar gas karbon dioksida dan perubahan grafik yang signifikan, dengan penyesuaian *input* data yang di terima oleh sensor MQ-2 dan dikirim oleh Mikrokontroler ESP8266 menuju *website*.

Pengguna dapat dengan jelas memantau tingkat intensitas gas CO<sub>2</sub>, mengamati status kualitas udara, mendapatkan saran yang sesuai, serta mengetahui efek dari paparan gas CO<sub>2</sub>. Selain itu, pengguna juga dapat memantau pola grafik dari perubahan kadar gas CO<sub>2</sub> secara visual.

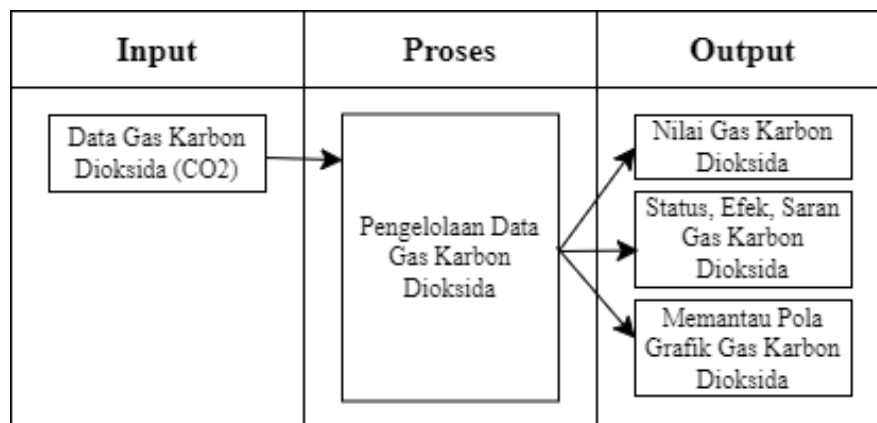
#### b) Blok Diagram

Blok diagram ini terdiri dari tiga tahap utama. Tahap pertama disebut "*input*", dimana sensor MQ-2 digunakan untuk mendeteksi keberadaan gas karbon dioksida. Dalam sistem ini, Mikrokontroler ESP8266 bertindak sebagai perangkat yang terhubung ke *internet* melalui jaringan Wi-Fi. Setelah terhubung, Mikrokontroler ESP8266 mengambil data dari sensor MQ-2 yang terus *me-monitor* gas karbon dioksida, informasi data tentang gas tersebut kemudian dikirimkan dari sensor menuju Mikrokontroler ESP8266.

Tahap kedua yaitu "*proses*". Pada tahap ini, data yang diterima dari sensor MQ-2 diolah untuk memantau tingkat gas karbon dioksida. Hasil pengolahan data ini kemudian dikirim ke program *Arduino IDE* untuk analisis lebih lanjut. Data yang sudah dianalisis ini kembali dikirim ke Mikrokontroler ESP8266. Di sini,

Mikrokontroler ESP8266 berperan penting dalam menampilkan data secara *realtime* pada *website* sederhana yang telah dibuat.

Tahap terakhir disebut "*output*", data yang telah diolah oleh Mikrokontroler ESP8266 akan ditampilkan dengan jelas pada *website* sederhana yang telah dibuat. Parameter utama yang ditampilkan adalah tingkat gas karbon dioksida. Jika tingkat gas tersebut rendah, *website* akan menampilkan status "aman" pada tabel status. Namun, jika tingkat gas karbon dioksida tinggi, *website* akan memberikan notifikasi "bahaya" pada tabel status. Hal ini bertujuan untuk memberikan informasi yang relevan tentang nilai gas karbon dioksida serta konsekuensi yang berkaitan dengan status dan dampaknya, seluruh tahap ini dapat dilihat dalam Gambar 4.2.

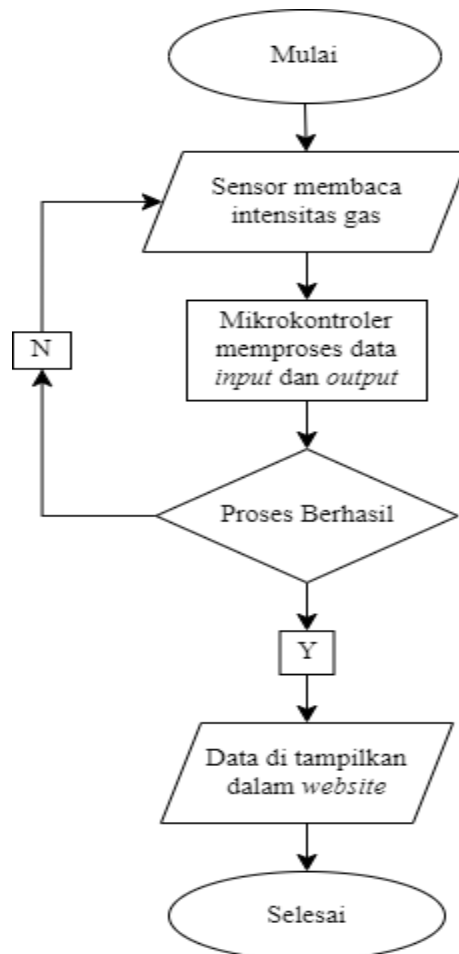


Gambar 4.2 Blok Diagram

Penjelasan daripada Gambar 4.2 berupa *input* data gas karbon dioksida yang diterima oleh sensor akan di proses melalui pengelolaan sistem *monitoring* pendeteksi kadar gas karbon dioksida. *Output* kadar gas karbon dioksida akan menunjukkan status “baik” jika nilai konsentrasi kadar gas karbon dioksida < 150 ppm, lalu akan menunjukkan status “sedang” jika nilai konsentrasi kadar gas karbon dioksida dari 151 - 250 ppm, lalu akan menunjukkan status “tidak sehat” jika nilai konsentrasi kadar gas karbon dioksida dari 251 - 350 ppm, lalu akan menunjukkan status “sangat tidak sehat” jika nilai konsentrasi kadar gas karbon dioksida dari 351 - 450 ppm, lalu akan menunjukkan status “berbahaya” jika nilai konsentrasi kadar gas karbon dioksida  $\geq$  451 ppm, dari Gambar 4.2 tersebut, terdapat fitur saran dari setiap status pada nilai kadar gas karbon dioksida tentang peringatan daripada nilai kadar gas karbon dioksida yang berubah sesuai dengan kondisi lapangan.

c) *Flowchart Sistem*

*Flowchart Sistem* digunakan sebagai alat untuk mendapatkan gambaran tentang tahapan dalam pembuatan sistem. Tampilan *flowchart sistem* yang menggambarkan proses pengambilan data, pemrosesan, dan penghasilan *output* dalam sistem dapat dilihat seperti yang ditunjukkan pada Gambar 4.3.

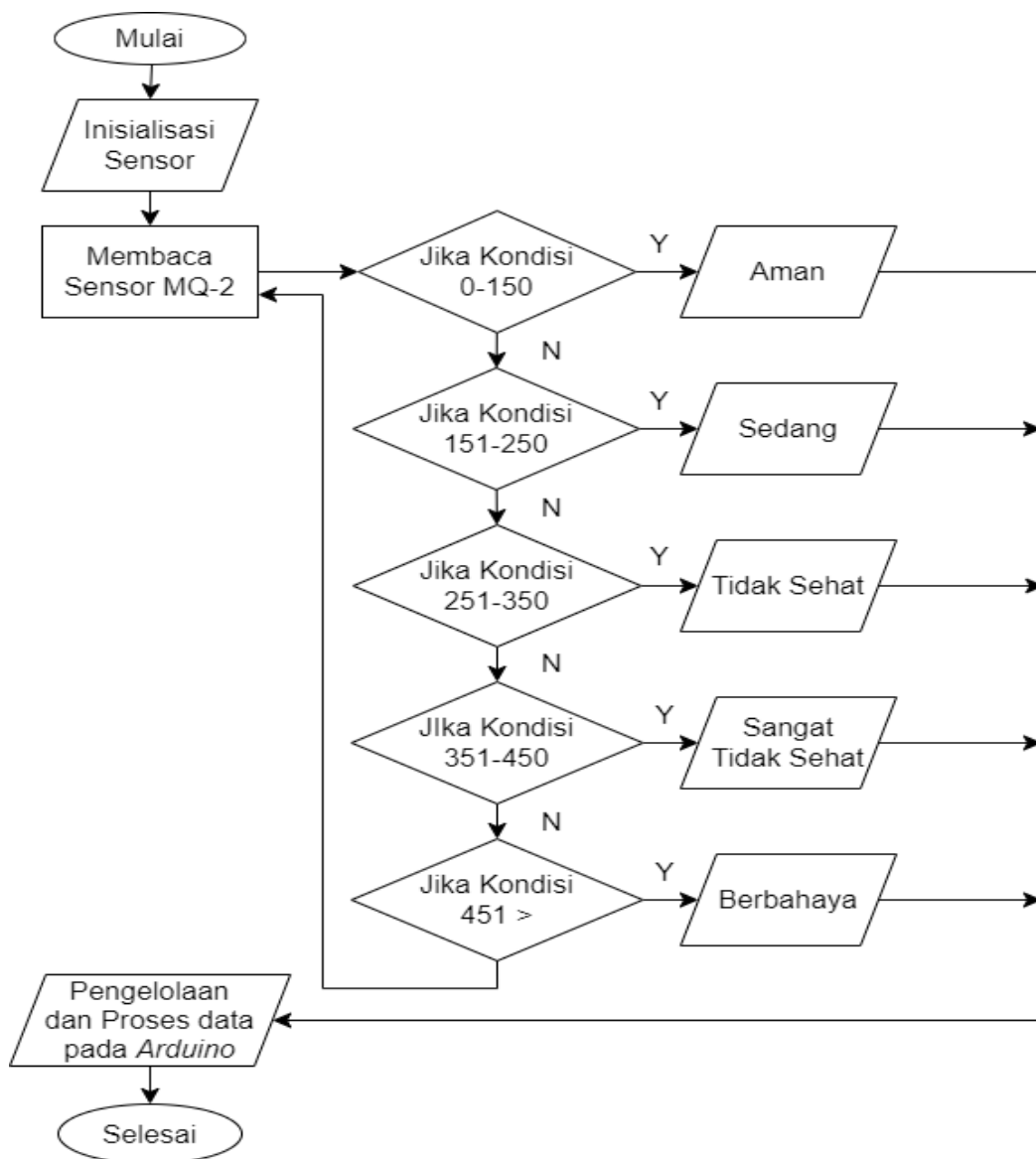


Gambar 4.3 *Flowchart Sistem*

Pertama, program akan menginisialisasi koneksi Wi-Fi dengan mode STA (*Station*). Jika Wi-Fi ditemukan, sistem akan segera mengumpulkan data melalui sensor. Data yang diperoleh dari sensor akan diproses oleh Mikrokontroler ESP8266 sebelum dikirimkan ke *Arduino IDE*. Mikrokontroler ESP8266 berperan sebagai pengirim data dari sensor dan bertugas untuk menampilkan sistem berdasarkan protokol yang telah dibuat dalam program ini. Selanjutnya, data yang diambil dari sensor akan ditampilkan dalam bentuk antarmuka pengguna pada sebuah *website* sederhana dengan keadaan *realtime*.

d) *Flowchart Rule Base Sensor*

*Flowchart Rule Base sensor* menjelaskan metode sensor dalam membaca intensitas dari gas karbon dioksida dan menghasilkan *output* pada *website* sederhana. Proses ini memberikan gambaran tentang bagaimana sensor bekerja untuk mendeteksi konsentrasi gas karbon dioksida dan mengolah data menjadi *output* yang relevan dan *realtime*, seperti yang ditunjukkan pada Gambar 4.4



Gambar 4.4 *Flowchart Rule Base Sensor*

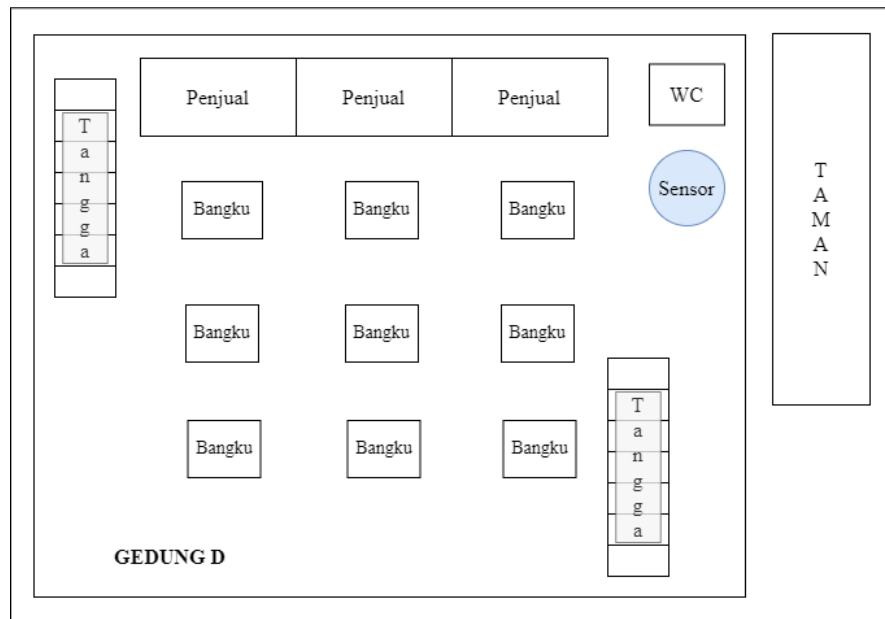
Proses yang telah ditunjukkan pada Gambar 4.4, menjelaskan bagaimana cara proses sensor bekerja yang pertama yaitu mulai kemudian sensor MQ-2 akan



mendeteksi intensitas gas karbon dioksida, apabila intensitas gas karbon dioksida yang didapat di bawah 150 ppm, maka status dalam *website* akan muncul “baik” dan akan muncul efek serta saran yang sesuai dengan kondisi status tersebut, apabila intensitas gas karbon dioksida yang didapat di bawah 250 ppm, maka status dalam *website* akan muncul “sedang” dan akan muncul efek serta saran yang sesuai dengan kondisi status tersebut, apabila intensitas gas karbon dioksida yang didapat di bawah 350 ppm, maka status dalam *website* akan muncul “tidak sehat” dan akan muncul efek serta saran yang sesuai dengan kondisi status tersebut, apabila intensitas gas karbon dioksida yang didapat di bawah 450 ppm, maka status dalam *website* akan muncul “sangat tidak sehat” dan akan muncul efek serta saran sesuai kondisi status tersebut, dan apabila intensitas gas karbon dioksida yang didapat di bawah 1000 ppm, maka status dalam *website* akan muncul “berbahaya” dan akan muncul efek dan saran yang sesuai dengan kondisi status tersebut.

e) Denah Alat

Langkah berikutnya yaitu pembuatan denah alat, yang bertujuan untuk menggambarkan letak serta penempatan alat-alat yang akan digunakan dalam penelitian tentang sistem pemantauan gas karbon dioksida yang direncanakan oleh penulis. Denah ini akan menyajikan visualisasi yang jelas mengenai lokasi strategis di mana alat-alat tersebut akan ditempatkan, dan rincian ini akan diilustrasikan dalam bentuk gambar denah yang dapat dilihat pada Gambar 4.5.

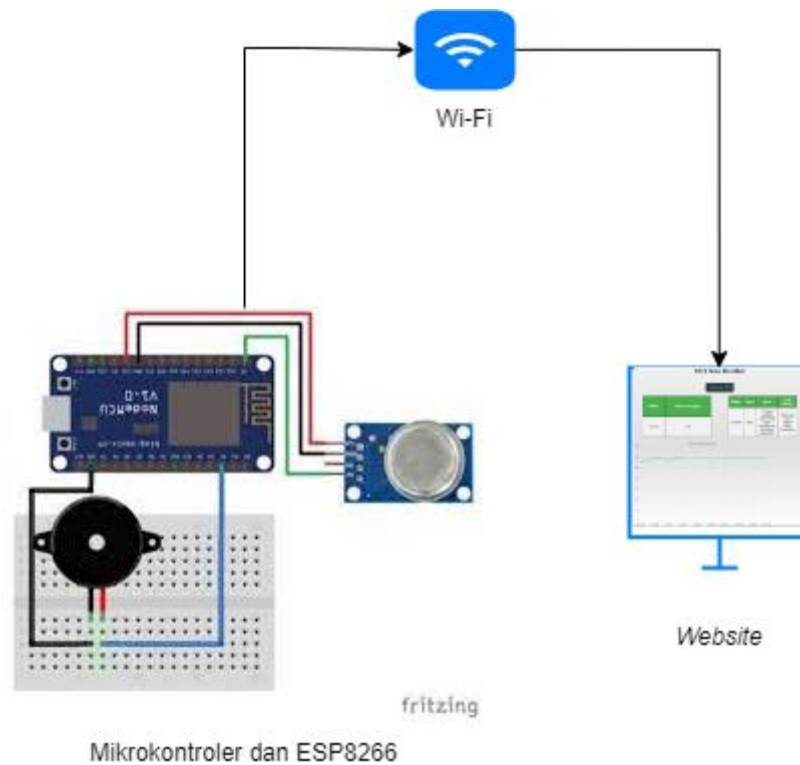


Gambar 4.5 Denah Alat

Pada Gambar 4.5, posisi dan penempatan alat yang akan digunakan untuk mengukur serta menganalisis konsentrasi gas karbon dioksida sangat jelas terlihat. Memilih lokasi yang optimal sangat penting untuk memastikan bahwa data yang diperoleh mencerminkan kondisi yang sebenarnya dan relevan dengan tujuan penelitian yang sedang dilakukan. Berdasarkan informasi yang digambarkan dalam Gambar 4.5, penulis berencana untuk menempatkan satu sensor di lokasi strategis pada Gedung D di Universitas Muhammadiyah Lamongan. Sensor akan ditempatkan berdekatan dengan kantin, dengan ketinggian sekitar 0,5 m dari permukaan tanah, yang diharapkan dapat memberikan informasi yang akurat mengenai kualitas udara di sekitar area tersebut melalui *website* sederhana yang dibangun.

#### f) Rancangan Alat

Proses perancangan alat melibatkan pembuatan rancangan untuk sebuah sensor pendeteksi gas karbon dioksida yaitu MQ-2 yang akan ditempatkan secara berdekatan dengan perangkat Mikrokontroler ESP8266. Tujuan dari alat tersebut adalah untuk mengumpulkan data dari sensor lalu dikirim dan diproses pada *Arduino IDE* dan mentransmisikannya melalui koneksi Wi-Fi ke sebuah *website* sederhana yang akan menampilkan informasi data kadar gas karbon dioksida tersebut, ilustrasi konsep ini dapat dilihat pada Gambar 4.6.



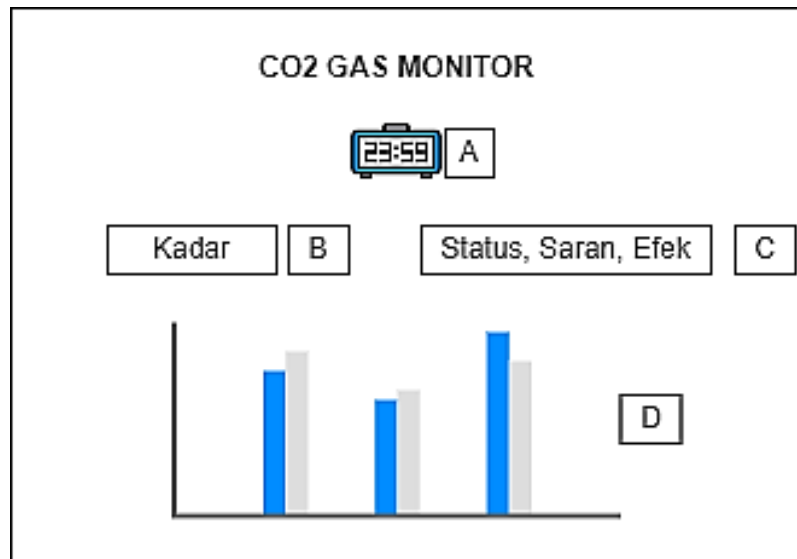
Gambar 4.6 Rancangan Alat

Penjelasan daripada Gambar 4.6 merupakan proses perancangan daripada alat penelitian tersebut, langkah pertama yang dilakukan yaitu menempatkan sensor MQ-2 secara strategis agar dapat mendeteksi dengan akurat konsentrasi gas karbon dioksida. Sensor ini kemudian dihubungkan dengan Mikrokontroler ESP8266, suatu mikrokontroler yang memiliki kemampuan untuk mengumpulkan data dari sensor tersebut dan mengatur komunikasi melalui jaringan Wi-Fi. Selanjutnya, setelah data gas karbon dioksida berhasil diambil oleh Mikrokontroler ESP8266, informasi akan dikirimkan melalui koneksi Wi-Fi menuju *server* lokal. Di sisi *server*, data tersebut akan diolah dan diproses untuk selanjutnya ditampilkan dalam tampilan *website* sederhana. Tampilan *website* ini berupa tabel nilai, tabel status dan tabel efek serta saran, dan tabel visual grafik yang relevan dengan konsentrasi gas karbon dioksida. Hasil dari perancangan alat ini dapat memberikan informasi yang berguna dalam mengawasi dan memantau kualitas gas karbon dioksida di sekitar lokasi perangkat berada. Dengan tampilan *website* sederhana, data yang terkumpul dapat dengan mudah diakses dan dimengerti oleh pengguna mengenai intensitas nilai gas karbon dioksida, dampak gas karbon dioksida dan

perubahannya, sehingga memberikan manfaat yang lebih luas dalam mengambil keputusan atau tindakan yang tepat terkait dengan kondisi gas karbon dioksida pada lingkungan sekitar.

g) Rancangan Antarmuka

Rancangan antarmuka memiliki peran yang sangat penting dalam mengilustrasikan struktur dan tata letak halaman pada sebuah situs *web*. Melalui rancangan antarmuka ini, pengguna dapat dengan lebih mudah membayangkan bagaimana elemen-elemen saling berinteraksi dalam konteks situs *web* tersebut. Dengan kata lain, rancangan antarmuka memberikan kemampuan kepada pengguna untuk lebih jelas memvisualisasikan hubungan antara berbagai elemen yang ada dalam situs *web*, sehingga memungkinkan pencapaian tujuan-tujuan yang diinginkan dengan lebih efektif. Dengan demikian, rancangan antarmuka bukan hanya tentang estetika visual, tetapi juga tentang menciptakan pengalaman pengguna yang intuitif dan memuaskan, proses ini di gambarkan dengan jelas pada Gambar 4.7.



Gambar 4.7 Rancangan Antarmuka

Berdasarkan pada Gambar 4.7, gambar tersebut menunjukkan 4 (empat) blok dengan penjelasan singkat dan keterangan sebagai berikut:

- a). Blok A bertujuan untuk mengetahui waktu *realtime* yang ditampilkan pada *website* dan telah terkalibrasi dengan waktu nyata.

- b). Blok B bertujuan untuk mengukur intensitas gas karbon dioksida yang terdeteksi oleh sensor MQ-2 dengan satuan ppm.
- c). Blok C menampilkan tabel status, saran dan efek dari kondisi gas karbon dioksida yang tertangkap oleh sensor dan memberikan informasi apakah kondisi udara di daerah tersebut dalam kondisi baik, sedang, tidak sehat, sangat tidak sehat dan berbahaya.
- d). Blok D berisi tabel visual grafik yang memungkinkan pengguna untuk melihat perubahan konsentrasi gas karbon dioksida secara *realtime*.

Dari 4 (empat) blok tersebut, penjelasan lebih lengkap mengenai 4 blok tersebut adalah sebagai berikut:

- a). Blok A berfungsi sebagai penunjuk waktu.
- b). Blok B berfungsi sebagai bagian yang menghubungkan sensor MQ-2 dengan sistem. Sensor MQ-2 memiliki tugas untuk mendeteksi tingkat gas karbon dioksida di sekitarnya dan mengukur intensitasnya. Data yang diperoleh dari sensor ini kemudian akan diproses lebih lanjut ke dalam sistem yang telah dirancang.
- c). Blok C bertanggung jawab untuk menampilkan status gas karbon dioksida. Berdasarkan data hasil sensor yang diolah, sistem akan menentukan apakah konsentrasi gas karbon dioksida masih dalam batas baik atau sudah mencapai tingkat bahaya. Informasi ini akan ditampilkan dalam bentuk status, saran dan efek sesuai kondisi gas karbon dioksida yang jelas dan mudah dipahami oleh pengguna.
- d). Blok D berperan dalam menampilkan representasi visual dari intensitas gas karbon dioksida yang diterima oleh sensor MQ-2. Fitur ini memungkinkan pengguna untuk secara *real-time* mengamati perubahan konsentrasi gas karbon. Dengan bantuan tampilan grafik yang disajikan dalam bentuk tabel visual, pengguna memiliki kemampuan untuk memonitor perubahan tingkat gas karbon dioksida dari waktu ke waktu secara langsung. Keberadaan grafik ini menjadi alat yang sangat berharga, karena memungkinkan pengguna untuk mengambil langkah tegas dengan cepat ketika terjadi fluktuasi tiba-tiba dalam konsentrasi gas karbon dioksida yang potensial mengkhawatirkan. Dengan

demikian, fasilitas ini dapat berperan dalam meningkatkan responsibilitas dan kesadaran terhadap lingkungan sekitar, sekaligus memberikan cara yang efektif untuk mengatasi situasi yang memerlukan tindakan segera terkait perubahan gas karbon dioksida yang terdeteksi.

#### 4.2.3 Implementasi

Implementasi merupakan aksi atau langkah-langkah yang diperlukan untuk menerapkan dan menjalankan suatu sistem, rencana, atau konsep. Ini melibatkan rangkaian tindakan yang telah direncanakan sebelumnya dan dilakukan dengan tujuan spesifik. Implementasi melibatkan langkah-langkah konkret yang diperlukan untuk mengubah suatu ide atau rencana menjadi kenyataan. Dalam konteks pengembangan sistem atau proyek, proses implementasi memastikan bahwa semua komponen atau fitur yang telah direncanakan diwujudkan dan berfungsi sesuai dengan harapan.

##### a) Implementasi Alat

Dalam menjalankan sistem *monitoring* gas karbon dioksida berbasis Mikrokontroler ESP8266 melalui sebuah *website* sederhana, diperlukan kelengkapan perangkat keras yang sesuai dengan kebutuhan. Untuk memberikan gambaran yang lebih jelas mengenai persyaratan yang diperlukan, berbagai komponen perangkat keras yang harus disiapkan guna mendukung kelancaran sistem ini telah diuraikan dalam Tabel 4.1.

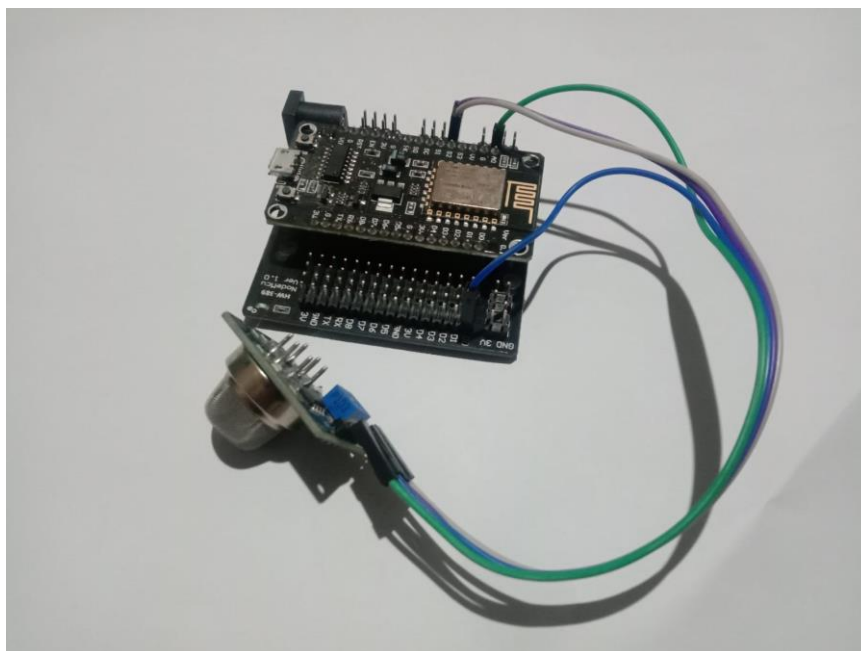
Tabel 4. 1 Komponen Perangkat Keras

| No | Perangkat Keras        | Fungsi   | Jumlah |
|----|------------------------|--|--------|
| 1. | Laptop                 | Menampilkan <i>Website</i> .   | 1      |
| 2. | Sensor MQ-2            | Pendeteksi Gas Karbon Dioksida.  | 1      |
| 3. | Mikrokontroler ESP8266 | Menyimpan data dan pengatur komunikasi Wi-Fi dengan modul Wi-Fi yang telah tersedia pada Mikrokontroler ESP8266. | 1      |
| 4. | USB <i>Micro Type</i>  | Media pemindahan <i>file</i> antar perangkat elektronik.   | 1      |
| 5. | <i>Buzzer</i>          | Sebagai perangkat notifikasi yang memunculkan suara.   | 1      |

Tabel 4.1 Komponen Perangkat Keras (lanjutan)

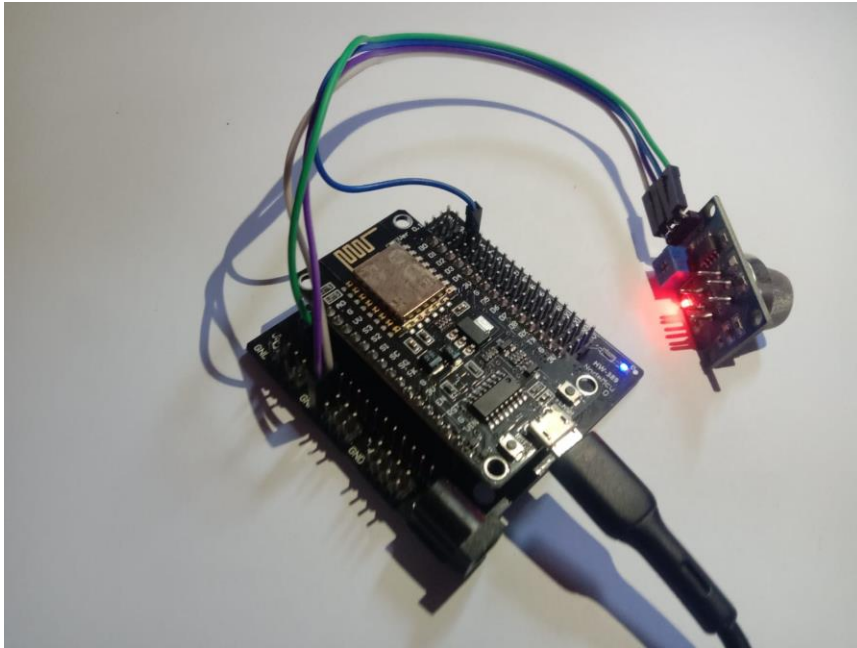
| No | Perangkat Keras     | Fungsi  | Jumlah |
|----|---------------------|---|--------|
| 6. | Kabel <i>Jumper</i> | Sebagai media penghubung antara sensor dan <i>buzzer</i> dengan NodeMCU ESP8266 | 6      |

Alat-alat tersebut telah dirancang sesuai dengan kebutuhan sistem *monitoring* gas karbon dioksida. Setelah menyelesaikan implementasi alat, gambaran hasil dari perangkaian sensor pendeteksi gas karbon MQ-2 dan Mikrokontroler ESP8266 dalam kondisi mati dapat dilihat pada Gambar 4.8.



Gambar 4.8 Sensor MQ-2 dan Mikrokontroler ESP8266 yang Mati

Kemudian, berikut merupakan gambaran dari sensor pendeteksi gas karbon dioksida MQ-2 bersama Mikrokontroler ESP8266 dalam kondisi aktif, perangkat ini telah siap untuk melakukan *monitoring* gas karbon dioksida, seperti yang ditunjukkan pada Gambar 4.9.

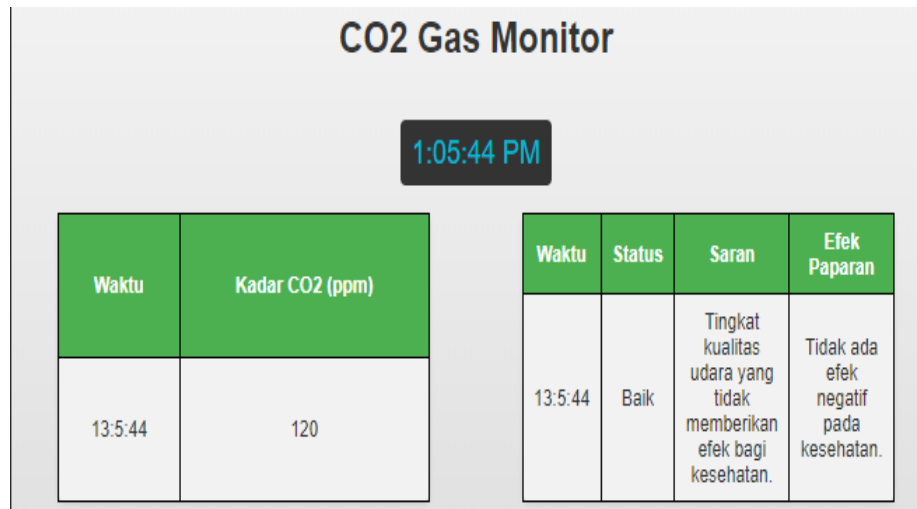


Gambar 4.9 Sensor MQ-2 dan Mikrokontroler ESP8266 yang Hidup

b) Implementasi Antarmuka

Pelaksanaan implementasi antarmuka diperlukan untuk mewujudkan desain yang telah direncanakan dalam konsep antarmuka menjadi kenyataan yang berfungsi secara optimal. Dalam fase implementasi, komponen-komponen desain antarmuka seperti susunan elemen, ikon, tombol, dan unsur interaksi lainnya akan diintegrasikan ke dalam sistem atau *platform* yang sedang digunakan. Proses ini memungkinkan pengguna untuk berinteraksi dengan situs *web* atau aplikasi dengan cara yang komprehensif dan efisien, ilustrasi mengenai pelaksanaan antarmuka dapat ditemukan dalam Gambar 4.10.

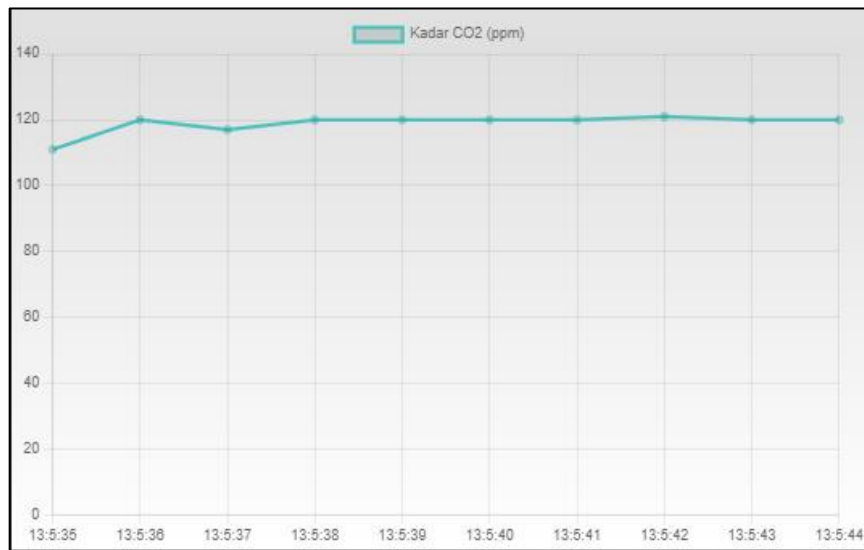




Gambar 4.10 Implementasi Antarmuka

Dilihat dari Gambar 4.10, implementasi antarmuka yang dihasilkan melalui penyusunan komponen-komponen pada perangkat lunak *Arduino IDE* menghasilkan data keluaran pada sebuah situs *web* sederhana. Situs ini menampilkan tabel berisi nilai ppm dari gas karbon dioksida, status kondisi gas karbon dioksida, rekomendasi yang disesuaikan dengan perubahan pada nilai gas karbon dioksida yang relevan, efek dari paparan karbon dioksida yang berdampak pada kesehatan masyarakat, serta visual grafik yang *realtime* terhadap perubahan kondisi yang diterima oleh sensor MQ-2.

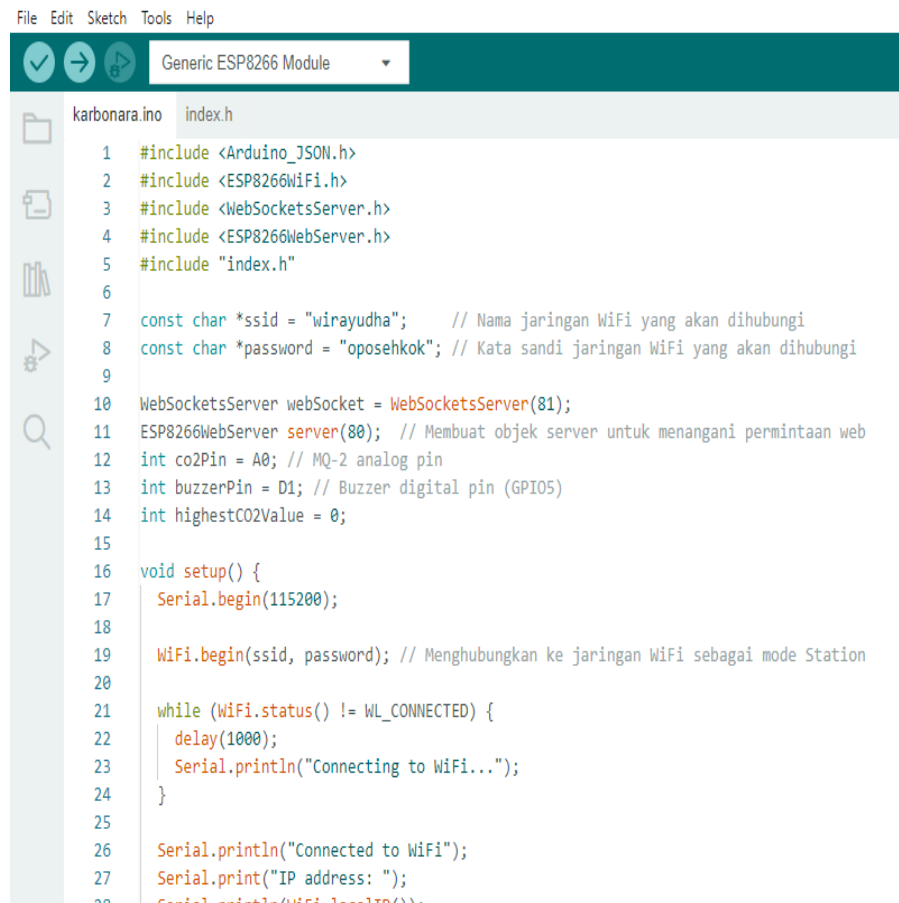
Sementara itu, mengenai visualisasi hasil dari implementasi antarmuka ini menampilkan sebuah gambaran grafik yang relevan dan terus diperbarui secara *realtime* sesuai dengan perubahan kondisi yang di *input* dari sensor MQ-2. Visualisasi grafik ini mampu memberikan gambaran jelas mengenai kualitas kinerja antarmuka dalam menyajikan *input* tampilan data dalam bentuk visual grafik daripada gas karbon dioksida secara aktual, gambaran hasil implementasi antarmuka secara detail dapat dilihat pada Gambar 4.11.



Gambar 4.11 Implementasi Antarmuka

Penjelasan mengenai Gambar 4.11 merupakan bentuk visualisasi dari perubahan kondisi berdasarkan nilai *input* gas karbon dioksida dalam bentuk visual grafik secara *realtime*. Grafik ini memiliki peran penting dalam memvisualisasikan perubahan kondisi pada gas karbon dioksida secara lebih aktual. Dengan adanya grafik ini, dapat disimpulkan bentuk perubahan pola yang mungkin terjadi dalam respons sistem terhadap perubahan nilai gas karbon dioksida. Grafik ini juga memiliki potensi untuk mendukung analisis dan evaluasi yang lebih holistik terhadap hasil penelitian. Selain itu, grafik ini dapat berperan sebagai opsi tambahan bagi masyarakat untuk memantau dan mengawasi perubahan-perubahan pada kadar gas karbon dioksida secara langsung, relevan, dan *realtime* dalam kondisi lapangan.

Adapun terkait implementasi antarmuka yang menunjukkan hasil dari usaha penulis dalam proses penulisan dan pengembangan kode pemrograman melalui *Arduino IDE* guna menciptakan sebuah situs *website* dengan tampilan sederhana, yang merupakan kelanjutan dari gambar visual yang telah ditampilkan pada Gambar 4.10 dan Gambar 4.11 sebelumnya, untuk lebih jelas mengenai penjelasan di atas, dapat dilihat pada Gambar 4.12.



```

File Edit Sketch Tools Help
Generic ESP8266 Module
karbonara.ino index.h
1 #include <Arduino_JSON.h>
2 #include <ESP8266WiFi.h>
3 #include <WebSocketsServer.h>
4 #include <ESP8266WebServer.h>
5 #include "index.h"
6
7 const char *ssid = "wirayudha"; // Nama jaringan WiFi yang akan dihubungi
8 const char *password = "oposehkok"; // Kata sandi jaringan WiFi yang akan dihubungi
9
10 WebSocketsServer webSocket = WebSocketsServer(81);
11 ESP8266WebServer server(80); // Membuat objek server untuk menangani permintaan web
12 int co2Pin = A0; // MQ-2 analog pin
13 int buzzerPin = D1; // Buzzer digital pin (GPIO5)
14 int highestCO2Value = 0;
15
16 void setup() {
17   Serial.begin(115200);
18
19   WiFi.begin(ssid, password); // Menghubungkan ke jaringan WiFi sebagai mode Station
20
21   while (WiFi.status() != WL_CONNECTED) {
22     delay(1000);
23     Serial.println("Connecting to WiFi...");
24   }
25
26   Serial.println("Connected to WiFi");
27   Serial.print("IP address: ");
28   Serial.println(WiFi.localIP());

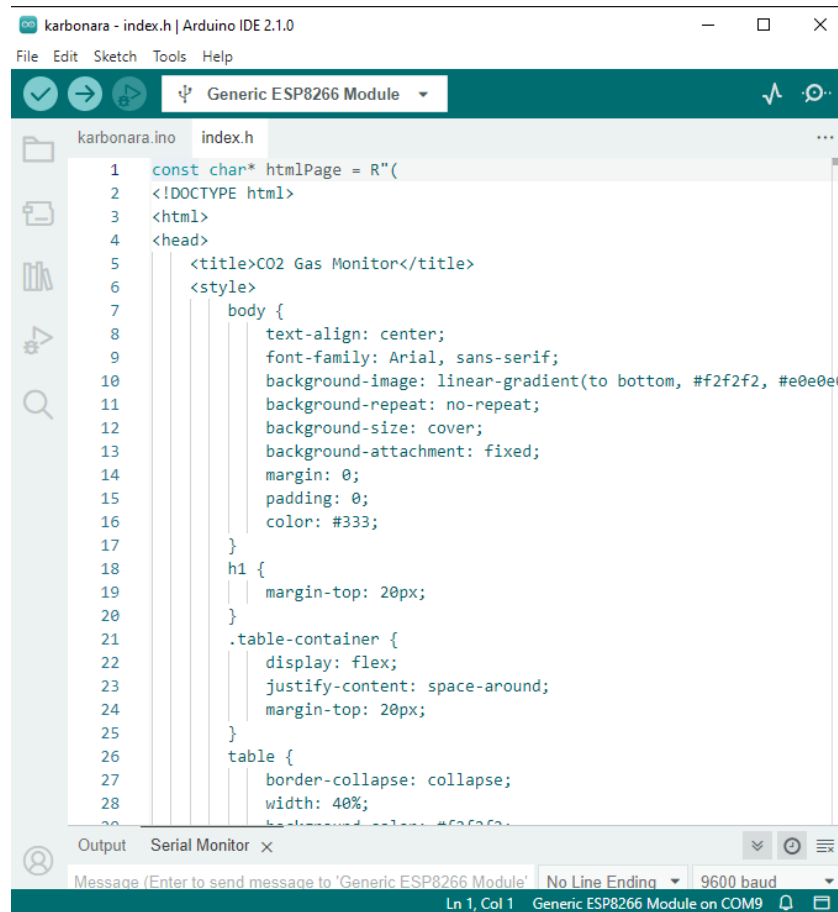
```

Gambar 4.12 Kode Implementasi Antarmuka

Pertama, dilihat dari Gambar 4.12, hasil implementasi antarmuka pemrograman menggunakan Bahasa Pemrograman C++ pada *platform Arduino IDE*. Perangkat lunak *Arduino IDE* digunakan untuk mempermudah proses pemrograman dan penerapan kode yang terlihat seperti pada gambar 4.12. Gambar tersebut secara jelas menggambarkan kode pemrograman yang dimanfaatkan untuk mengendalikan sensor pada perangkat menggunakan Bahasa C++ melalui *Arduino IDE*. Penulis juga menyertakan kode untuk sistem berbasis aturan (*rule based*) yang digunakan untuk mengkalibrasi hasil data yang diperoleh dari sensor MQ-2. Data ini diintegrasikan dengan situs *web* sistem *monitoring* gas karbon dioksida yang sesuai.

Kedua, untuk hasil kode pemrograman daripada *website* yang di program menggunakan Bahasa Pemrograman HTML guna untuk menampilkan *input* data sensor yang divisualisasikan dengan nilai kadar gas karbon dioksida ke dalam

bentuk tabel dan visual grafik yang sederhana, relevan dan *realtime* pada Bahasa Pemrograman HTML tersebut, dapat dilihat pada Gambar 4.13.



```

karbonara - index.h | Arduino IDE 2.1.0
File Edit Sketch Tools Help
Generic ESP8266 Module
karbonara.ino index.h
1 const char* htmlPage = R"(
2 <!DOCTYPE html>
3 <html>
4 <head>
5   <title>CO2 Gas Monitor</title>
6   <style>
7     body {
8       text-align: center;
9       font-family: Arial, sans-serif;
10      background-image: linear-gradient(to bottom, #f2f2f2, #e0e0e0);
11      background-repeat: no-repeat;
12      background-size: cover;
13      background-attachment: fixed;
14      margin: 0;
15      padding: 0;
16      color: #333;
17    }
18    h1 {
19      margin-top: 20px;
20    }
21    .table-container {
22      display: flex;
23      justify-content: space-around;
24      margin-top: 20px;
25    }
26    table {
27      border-collapse: collapse;
28      width: 40%;
29      background-color: #f2f2f2;

```

Gambar 4.13 Kode Implementasi Antarmuka

Berdasarkan Gambar 4.13, menunjukkan sebuah hasil penulisan kode pemrograman dari tampilan *website* menggunakan Bahasa Pemrograman HTML yang merupakan kode pemrograman dasar pada *Arduino IDE* untuk menghasilkan sebuah *website* dengan tampilan yang sederhana. Contoh tersebut merupakan penggambaran tentang cara efektif dalam mengintegrasikan elemen-elemen dalam halaman *website*. Proses pembuatan tampilan *website* dengan Bahasa Pemrograman HTML ini memberikan kemudahan dalam metode pengembangan pada penelitian lebih lanjut dan memastikan tampilan halaman *website* tetap konsisten dalam penggunaan.

## 4.2. Pengujian

Pengujian memegang peranan penting dalam proses evaluasi dan pemeriksaan sistem atau teori secara terencana dan terkendali. Tujuannya adalah untuk menguji serta memvalidasi keefektifan dan keabsahan kinerja sistem tersebut. Melalui tahap ini, sistem atau teori akan diuji dalam berbagai skenario untuk mengidentifikasi potensi kelemahan, memastikan bahwa semua komponen berfungsi sesuai yang diharapkan, dan memastikan bahwa hasil yang diperoleh dapat diandalkan. Dengan melakukan pengujian yang cermat, kita dapat memastikan bahwa sistem atau teori tersebut siap untuk diimplementasikan secara lebih luas atau dapat digunakan sebagai dasar yang kuat dalam pengambilan keputusan.

### 4.2.1 Analisa Sistem

Pada tahap analisis sistem ini, berbagai aspek dan fitur yang ada dalam sistem akan dievaluasi. Hasil dari pengujian tersebut akan dianalisis secara mendalam untuk melihat dan menilai performa, serta kinerja dari sistem secara keseluruhan, informasi yang diperoleh dari hasil pengujian sistem dan perangkat dapat dilihat pada Tabel 4.2.

Tabel 4.2 Jarak sensor MQ-2 Sejauh 10 cm dari Sumber Gas Karbon Dioksida

| No | Jarak Maksimal Sensor | Waktu    | Kadar | Status | Saran/Keterangan   | Efek   |
|----|-----------------------|----------|-------|--------|--|--|
| 1. | 10 cm                 | 09:10:35 | 205   | SEDANG | Kondisi udara cukup Sedang, Tingkat kualitas udara yang tidak memberikan efek bagi kesehatan manusia atau hewan, tetapi berpengaruh pada tumbuhan yang sensitive dan nilai estetika. | Mungkin menyebabkan sedikit ketidaknyamanan pada beberapa orang. |

Tabel 4.2 Jarak sensor MQ-2 Sejauh 10 cm dari Sumber Gas Karbon Dioksida (Lanjutan)

| No. | Jarak Maksimal Sensor | Waktu    | Kadar | Status         | Saran/Keterangan  | Efek   |
|-----|-----------------------|----------|-------|----------------|---|--|
| 2.  | 10 cm                 | 09:10:36 | 230   | SEDA<br>NG     | Kondisi udara cukup Sedang, Tingkat kualitas udara yang tidak memberikan efek bagi kesehatan manusia atau hewan, tetapi berpengaruh pada tumbuhan yang sensitive dan nilai estetika.              | Mungkin menyebabkan sedikit ketidaknyamanan pada beberapa orang.                   |
| 3.  | 10 cm                 | 09:10:37 | 246   | SEDA<br>NG     | Kondisi udara cukup Sedang, Tingkat kualitas udara yang tidak memberikan efek bagi kesehatan manusia atau hewan, tetapi berpengaruh pada tumbuhan yang sensitive dan nilai estetika.              | Mungkin menyebabkan sedikit ketidaknyamanan pada beberapa orang.                   |
| 4.  | 10 cm                 | 09:10:38 | 287   | TIDAK<br>SEHAT | Kondisi udara Tidak Sehat, Tingkat kualitas udara yang bersifat merugikan pada manusia ataupun kelompok hewan yang sensitif atau bisa menimbulkan kerusakan pada tumbuhan ataupun nilai estetika. | Paparan berkepanjangan dapat menyebabkan sakit kepala, pusing, dan gejala lainnya. |

Tabel 4.2 Jarak sensor MQ-2 Sejauh 10 cm dari Sumber Gas Karbon Dioksida (Lanjutan)

| No | Jarak Maksimal Sensor | Waktu    | Kadar | Status      | Saran/Keterangan  | Efek   |
|----|-----------------------|----------|-------|-------------|---|--|
| 5. | 10 cm                 | 09:10:39 | 336   | TIDAK SEHAT | Kondisi udara Tidak Sehat, Tingkat kualitas udara yang bersifat merugikan pada manusia ataupun kelompok hewan yang sensitif atau bisa menimbulkan kerusakan pada tumbuhan ataupun nilai estetika. | Paparan berkepanjangan dapat menyebabkan sakit kepala, pusing, dan gejala lainnya. |
| 6. | 10 cm                 | 09:10:40 | 340   | TIDAK SEHAT | Kondisi udara Tidak Sehat, Tingkat kualitas udara yang bersifat merugikan pada manusia ataupun kelompok hewan yang sensitif atau bisa menimbulkan kerusakan pada tumbuhan ataupun nilai estetika. | Paparan berkepanjangan dapat menyebabkan sakit kepala, pusing, dan gejala lainnya. |
| 7. | 10 cm                 | 09:10:41 | 323   | TIDAK SEHAT | Kondisi udara Tidak Sehat, Tingkat kualitas udara yang bersifat merugikan pada manusia ataupun kelompok hewan yang sensitif atau bisa menimbulkan kerusakan pada tumbuhan ataupun nilai estetika. | Paparan berkepanjangan dapat menyebabkan sakit kepala, pusing, dan gejala lainnya. |

Tabel 4.2 Jarak sensor MQ-2 Sejauh 10 cm dari Sumber Gas Karbon Dioksida (Lanjutan)

| No  | Jarak Maksimal Sensor | Waktu    | Kadar | Status      | Saran/Keterangan  | Efek   |
|-----|-----------------------|----------|-------|-------------|---|--|
| 8.  | 10 cm                 | 09:10:42 | 298   | TIDAK SEHAT | Kondisi udara Tidak Sehat, Tingkat kualitas udara yang bersifat merugikan pada manusia ataupun kelompok hewan yang sensitif atau bisa menimbulkan kerusakan pada tumbuhan ataupun nilai estetika. | Paparan berkepanjangan dapat menyebabkan sakit kepala, pusing, dan gejala lainnya. |
| 9.  | 10 cm                 | 09:10:43 | 251   | TIDAK SEHAT | Kondisi udara Tidak Sehat, Tingkat kualitas udara yang bersifat merugikan pada manusia ataupun kelompok hewan yang sensitif atau bisa menimbulkan kerusakan pada tumbuhan ataupun nilai estetika. | Paparan berkepanjangan dapat menyebabkan sakit kepala, pusing, dan gejala lainnya. |
| 10. | 10 cm                 | 09:10:44 | 227   | SEDANG      | Kondisi udara cukup Sedang, Tingkat kualitas udara yang tidak memberikan efek bagi kesehatan manusia atau hewan, tetapi berpengaruh pada tumbuhan yang sensitive dan nilai estetika.              | Mungkin menyebabkan sedikit ketidaknyamanan pada beberapa orang.                   |



Tabel 4.2 Jarak sensor MQ-2 Sejauh 10 cm dari Sumber Gas Karbon Dioksida (Lanjutan)

| No  | Jarak Maksimal Sensor | Waktu    | Kadar | Status | Saran/Keterangan   | Efek   |
|-----|-----------------------|----------|-------|--------|--|--|
| 11. | 10 cm                 | 09:10:45 | 210   | SEDANG | Kondisi udara cukup Sedang, Tingkat kualitas udara yang tidak memberikan efek bagi kesehatan manusia atau hewan, tetapi berpengaruh pada tumbuhan yang sensitive dan nilai estetika. | Mungkin menyebabkan sedikit ketidaknyamanan pada beberapa orang. |
| 12. | 10 cm                 | 09:10:46 | 203   | SEDANG | Kondisi udara cukup Sedang, Tingkat kualitas udara yang tidak memberikan efek bagi kesehatan manusia atau hewan, tetapi berpengaruh pada tumbuhan yang sensitive dan nilai estetika. | Mungkin menyebabkan sedikit ketidaknyamanan pada beberapa orang. |
| 13. | 10 cm                 | 09:10:47 | 198   | SEDANG | Kondisi udara cukup Sedang, Tingkat kualitas udara yang tidak memberikan efek bagi kesehatan manusia atau hewan, tetapi berpengaruh pada tumbuhan yang sensitive dan nilai estetika. | Mungkin menyebabkan sedikit ketidaknyamanan pada beberapa orang. |

Tabel 4.2 Jarak sensor MQ-2 Sejauh 10 cm dari Sumber Gas Karbon Dioksida (Lanjutan)

| No  | Jarak Maksimal Sensor | Waktu    | Kadar | Status | Saran/Keterangan   | Efek   |
|-----|-----------------------|----------|-------|--------|--|--|
| 14. | 10 cm                 | 09:10:48 | 174   | SEDANG | Kondisi udara cukup Sedang, Tingkat kualitas udara yang tidak memberikan efek bagi kesehatan manusia atau hewan, tetapi berpengaruh pada tumbuhan yang sensitive dan nilai estetika. | Mungkin menyebabkan sedikit ketidaknyamanan pada beberapa orang. |
| 15. | 10 cm                 | 09:10:49 | 162   | SEDANG | Kondisi udara cukup Sedang, Tingkat kualitas udara yang tidak memberikan efek bagi kesehatan manusia atau hewan, tetapi berpengaruh pada tumbuhan yang sensitive dan nilai estetika. | Mungkin menyebabkan sedikit ketidaknyamanan pada beberapa orang. |
| 16. | 10 cm                 | 09:10:50 | 153   | SEDANG | Kondisi udara cukup Sedang, Tingkat kualitas udara yang tidak memberikan efek bagi kesehatan manusia atau hewan, tetapi berpengaruh pada tumbuhan yang sensitive dan nilai estetika. | Mungkin menyebabkan sedikit ketidaknyamanan pada beberapa orang. |

Tabel 4.2 Jarak sensor MQ-2 Sejauh 10 cm dari Sumber Gas Karbon Dioksida (Lanjutan)

| No  | Jarak Maksimal Sensor | Waktu    | Kadar | Status | Saran/Keterangan  | Efek                                   |
|-----|-----------------------|----------|-------|--------|---|--|
| 17. | 10 cm                 | 09:10:51 | 148   | BAIK   | Tingkat kualitas udara yang tidak memberikan efek bagi kesehatan. | Tidak ada efek negatif pada kesehatan. |
| 18. | 10 cm                 | 09:10:52 | 144   | BAIK   | Tingkat kualitas udara yang tidak memberikan efek bagi kesehatan. | Tidak ada efek negatif pada kesehatan. |
| 19. | 10 cm                 | 09:10:53 | 147   | BAIK   | Tingkat kualitas udara yang tidak memberikan efek bagi kesehatan. | Tidak ada efek negatif pada kesehatan. |
| 20. | 10 cm                 | 09:10:54 | 146   | BAIK   | Tingkat kualitas udara yang tidak memberikan efek bagi kesehatan. | Tidak ada efek negatif pada kesehatan. |

Berdasarkan data yang terdapat pada Tabel 4.2 tersebut dapat diamati bahwa, ketika sensor gas karbon dioksida MQ-2 ditempatkan pada jarak 10 cm, menghasilkan *output* yang secara *realtime* menampilkan urutan data gas karbon dioksida yang di *input* oleh sensor MQ-2 setiap 1 detik sekali. Hasil tersebut menunjukkan bahwa sensor MQ-2 mampu secara akurat dan berkala dalam mendeteksi perubahan terhadap konsentrasi dari kadar gas karbon dioksida pada jarak yang telah ditentukan dalam penelitian ini. Mengenai data yang telah dikumpulkan pada tabel 4.2, terdapat 3 status yang terdeteksi dan menyebabkan alarm menyala 6 kali, berikut klasifikasi dari himpunan data dari tabel tersebut, dapat dilihat pada Tabel 4.3.

Tabel 4.3 Jumlah data yang diambil dari pengujian di Gedung D

| No. | Status             | Terdeteksi | Alarm Menyala |
|-----|--------------------|------------|---------------|
| 1.  | Baik               | 4 kali     | 0 kali        |
| 2.  | Sedang             | 10 kali    | 0 kali        |
| 3.  | Tidak Sehat        | 6 kali     | 6 kali        |
| 4.  | Sangat Tidak Sehat | 0 kali     | 0 kali        |
| 5.  | Berbahaya          | 0 kali     | 0 kali        |

Dari 20 data yang diambil pada tabel 4.2 tersebut, dapat dilihat kualitas udara dengan status “Baik” mencapai sekitar 20% dari data keseluruhan, kemudian kualitas udara dengan status “Sedang” mencapai sekitar 50% dari data keseluruhan. Kemudian kualitas udara dengan status “Tidak Sehat” mencapai sekitar 30% dari data keseluruhan, namun kualitas udara dengan status “Sangat Tidak Sehat” dan “Berbahaya” tidak terdeteksi pada sensor dengan presentase sekitar 0%. Dari data tersebut, dapat dilihat mengenai kualitas udara dengan status “Tidak Sehat” menyebabkan *alarm* menyala sebanyak 6 kali dikarenakan udara dengan status “Tidak Sehat” terdeteksi sebanyak 6 kali dari data yang dihimpun oleh sensor pendeteksi gas karbon dioksida.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan hasil yang telah diperoleh dari penelitian dalam pembuatan Sistem *Monitoring* Kadar Gas Karbon Dioksida (CO<sub>2</sub>) Berbasis *Internet Of Things*, dapat disimpulkan bahwa:

1. Sistem *monitoring* gas karbon dioksida dengan menggunakan *website* telah dibangun dan berhasil diimplementasikan dengan hasil yang relevan terhadap kondisi pada kantin di Gedung D pada Universitas Muhammadiyah Lamongan, sistem ini dibangun menggunakan Bahasa Pemrograman C++ dan HTML agar menghasilkan tampilan *website* yang sederhana dan *user friendly* bagi pengguna.
2. Sistem ini berhasil dibuat oleh peneliti guna untuk memantau dan mengawasi kondisi kadar gas karbon dioksida pada kantin di Gedung D pada Universitas Muhammadiyah Lamongan dengan menggunakan metode *rulebase* guna untuk menentukan batas-batas nilai kadar gas karbon dioksida dengan tujuan untuk mengawasi perubahan-perubahan kadar gas secara relevan dan *realtime*.

#### **5.2 Saran**

Hasil dari penelitian ini memberikan beberapa saran yang dapat di terapkan pada penelitian selanjutnya, antara lain:

1. Penelitian ini hanya memanfaatkan 1 (satu) sensor saja, khususnya sensor gas karbon dioksida MQ-2. Oleh karena itu, untuk penelitian mendatang, disarankan agar lebih dari satu sensor digunakan dalam rangka meningkatkan akurasi dan cakupan pengukuran. Dengan mengintegrasikan beberapa sensor gas karbon dioksida MQ-2, penelitian selanjutnya dapat menghasilkan hasil yang lebih komprehensif dan dapat diandalkan dalam memantau tingkat gas karbon dioksida di lingkungan yang berbeda.
2. Penelitian ini tidak menggunakan *database* sebagai tempat menyimpan data *realtime*-nya, sebab penulis hanya berfokus pada penelitian berupa perubahan kadar gas karbon dioksida saja, sehingga diharapkan untuk penelitian

selanjutnya dapat menggunakan *database* guna mengetahui riwayat data kadar gas karbon dioksida sebelumnya.

3. Penelitian ini hanya menggunakan sensor gas karbon dioksida tanpa perangkat keras tambahan, sehingga diharapkan untuk penelitian selanjutnya dapat menambahkan perangkat keras lainnya yang sesuai dengan fokus dari pengembangan terhadap penelitian selanjutnya berdasarkan penelitian ini.
4. Penelitian ini belum menerapkan metode *scientific*, sehingga pada penelitian selanjutnya, diharapkan adanya penerapan metode *scientific* guna untuk mendapatkan hasil yang lebih akurat dan dapat diandalkan. Dengan demikian, keberhasilan dan validitas penelitian dapat ditingkatkan secara signifikan, menjadikan temuan yang diperoleh lebih relevan terhadap pembuatan sistem ini.

## DAFTAR PUSTAKA

- Aji Dwi Komara, E. C. (2016). Sistem Pendukung Keputusan Penentuan Prioritas Pemadaman Hotspot Kebakaran Hutan Dan Lahan Menggunakan Metode Analytic Hierarchy Process Dan Weighted Product. 2.
- Amsar Dkk. (2020). Perancangan Alat Pendeteksi Co2 Menggunakan Sensor Mq-2 Berbasis Internet Of Thing. *Perancangan Alat Pendeteksi Co2 Menggunakan Sensor Mq-2 Berbasis Internet Of Thing*.
- Aprillya Dan Chasanah. (2022). Sistem Pendukung Keputusan Identifikasi Daerah Rawan Kekeringandengan Metode Fuzzy Analytical Hierarchy Process (Studi Kasus : Kabupaten Lamongan). 3(2).
- Ardiansyah. (2020). Deteksi Jarak Kedatangan Kereta Api Padapersimpangan Jalan Menggunakan Sensor Accelerometer Dan Neural Network. *Deteksi Jarak Kedatangan Kereta Api Padapersimpangan Jalan Menggunakan Sensor Accelerometer Dan Neural Network*, 43.
- Asteroid Dkk. (2016). Analisis Wireless Local Area Network (WLAN) Dan Perancangan MAC Address Filtering Menggunakan Mikrotik. *Analisis Wireless Local Area Network (WLAN) Dan Perancangan MAC Address Filtering Menggunakan Mikrotik*.
- Ayu Dkk. (2020). *Sistem Monitoring Kegiatan Akademik Siswa. Sistem Monitoring Kegiatan Akademik Siswa*.
- Bianto. (2019). Perancangan Sistem Klasifikasi Penyakit Jantung. *Perancangan Sistem Klasifikasi Penyakit Jantung*, 76.
- Cahyanto Dkk. (2020). Pengujian Rule-Based Pada Dataset Log Server Menggunakan Support Vector Machine Berbasis Linear Discriminat Analysis Untuk Deteksi Malicious Activity. *Pengujian Rule-Based Pada Dataset Log Server Menggunakan Support Vector Machine Berbasis Linear Discriminat Analysis Untuk Deteksi Malicious Activity*.
- Chrisdianta Dkk. (2020). Sistem Pendukung Keputusan Penentuan Pemesanan Barang Menggunakan Logika Fuzzy
- Tsukamoto. *Sistem Pendukung Keputusan Penentuan Pemesanan Barang Menggunakan Logika Fuzzy Tsukamoto*.

- Efendi, Y. (2018). Internet Of Things (IOT) Sistem Pengendalian Lampu Menggunakan Raspberry Pi Berbasis Mobile. *Jurnal Ilmiah Ilmu Komputer*, 4(2), 21–27.
- Febriani, W. D. (2023). Perancangan Sistem Informasi Akuntansipersediaan Barang Pada Sekretariatdprd Kota Palembang. *Perancangan Sistem Informasi Akuntansipersediaan Barang Pada Sekretariatdprd Kota Palembang*, 106.
- Handoyo Dkk. (2019). Jurnal Sistem Informasi Bisnis. *Analisis Keamanan Sistem Informasi Berdasarkan Framework COBIT 5 Menggunakan Capability Maturity*, 47-48.
- Imansyah, F. (2021). Sistem Informasi Geografis Lahan Pertanian Rawan Kebakaran. 9(2).
- Khoiruddin. (2018). Sistem Monitoring Gas Co2 Di Gunung. *Sistem Monitoring Gas Co2 Di Gunung*.
- Mahendra Dan Sasmoko. (2017). Rancang Bangun Sistem Pendeteksi Kebakaran Berbasis Iot Dan SMS Gateway. *Rancang Bangun Sistem Pendeteksi Kebakaran Berbasis Iot Dan SMS Gateway*.
- Neyfa Dan Tamara. (2016). Special Meeting Of Council--Private Practice And Juniors' Contract. *Special Meeting Of Council--Private Practice And Juniors' Contract*.
- Pangestu Dkk. (2019). Sistem Monitoring Beban Listrik Berbasis Arduino Nodemcu Esp8266. *Sistem Monitoring Beban Listrik Berbasis Arduino Nodemcu Esp8266*.
- Pratama Dkk. (2020). Perancangan Sistem Monitoring Ketinggian Air Sebagai Pendeteksi Banjir Berbasis Iot Menggunakan Sensor Ultrasonik. *Perancangan Sistem Monitoring Ketinggian Air Sebagai Pendeteksi Banjir Berbasis Iot Menggunakan Sensor Ultrasonik*.
- Rasyid, F. (2014). Permasalahan Dan Dampak Kebakaran Hutan. *Edisi 1*(4).
- Rian Nugraha, D. G. (2022). Sistem Pendukung Keputusan Penerimaan Bantuan Sosial Dengan Metode Fuzzy Analytical Hierarchy Process. *11*(P-ISSN 2301-7988, E-ISSN 2581-0588).



- Saputra Dan Shodiq. (2022). Grey Forecasting Model Untuk Peramalan Harga Ikan Budidaya. *Grey Forecasting Model Untuk Peramalan Harga Ikan Budidaya*.
- Setiadi Dkk. (2018). Penerapan Internet Of Things (Iot) Pada Sistem Monitoring Irigasi. *Penerapan Internet Of Things (Iot) Pada Sistem Monitoring Irigasi*.
- Sucipto Dkk. (2023). Rancang Bangun Alat Deteksi Dan Pengukur Gas Emisi Karbondioksida (CO<sub>2</sub>) Dan. *Rancang Bangun Alat Deteksi Dan Pengukur Gas Emisi Karbondioksida (CO<sub>2</sub>) Dan*.
- Sumardi Dan Mulyati. (2018). Internet Of Things (Iot) Pada Prototipe Pendeteksi Kebocoran Gas Berbasis Mq-2 Dan Sim800l. *Internet Of Things (Iot) Pada Prototipe Pendeteksi Kebocoran Gas Berbasis Mq-2 Dan Sim800l*.
- Susilo. (2018). Rancang Bangun Website Toko Online Menggunakan Metode Waterfall. . *Rancang Bangun Website Toko Online Menggunakan Metode Waterfall. .*

## LAMPIRAN

### Lampiran 1. Code System (Sketch Arduino IDE)

```
#include <Arduino_JSON.h>
#include <ESP8266WiFi.h>
#include <WebSocketsServer.h>
#include <ESP8266WebServer.h>
#include "index.h"

const char *ssid = "wirayudha"; // Nama jaringan
WiFi yang akan dihubungi
const char *password = "oposehkok"; // Kata sandi
jaringan WiFi yang akan dihubungi

WebSocketsServer webSocket = WebSocketsServer(81);
ESP8266WebServer server(80); // Membuat objek
server untuk menangani permintaan web
int co2Pin = A0; // MQ-2 analog pin
int buzzerPin = D1; // Buzzer digital pin (GPIO5)
int highestCO2Value = 0;

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

  WiFi.begin(ssid, password); // Menghubungkan ke
jaringan WiFi sebagai mode Station

  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }

  Serial.println("Connected to WiFi");
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());

  webSocket.begin();

  pinMode(co2Pin, INPUT);
  pinMode(buzzerPin, OUTPUT);

  digitalWrite(buzzerPin, LOW); // Matikan buzzer
saat startup
```

```
    server.on("/", HTTP_GET, handleRoot); //
Menangani permintaan ke root URL ("/")
    server.begin();
}

void loop() {
    websocket.loop();
    server.handleClient();

    int co2Value = analogRead(co2Pin);

    if (co2Value > highestCO2Value) {
        highestCO2Value = co2Value;
    }

    String status = getCO2Status(co2Value);
    String advice = getCO2Advice(status);
    String effect = getCO2Effect(status);

    JSONVar json;
    json["co2"] = co2Value;
    json["status"] = status;
    json["advice"] = advice;
    json["effect"] = effect;

    String jsonString = JSON.stringify(json);

    websocket.broadcastTXT(jsonString);
    if (co2Value == highestCO2Value) {
        digitalWrite(buzzerPin, HIGH); // Aktifkan
buzzer saat nilai tertinggi tercapai
    } else {
        digitalWrite(buzzerPin, LOW); // Matikan buzzer
jika tidak ada nilai tertinggi
    }

    delay(1000); // Delay antara pembacaan
}
```

```
String getCO2Status(int co2Value) {
    if (co2Value < 150) {
        return "Baik";
    } else if (co2Value < 250) {
        return "Sedang";
    } else if (co2Value < 350) {
        return "Tidak Sehat";
    } else if (co2Value < 450) {
        return "Sangat Tidak Sehat";
    } else (co2Value > 450) {
        return "Berbahaya";
    }
}

String getCO2Advice(String status) {
    if (status == "Baik") {
        return "Kondisi udara bagus, tetap jaga
kebersihan lingkungan.";
    } else if (status == "Sedang") {
        return "Kondisi udara cukup baik, pastikan ada
ventilasi yang memadai.";
    } else if (status == "Tidak Sehat") {
        return "Kondisi gas karbon dioksida mulai
meluas, segera ambil tindakan yang aman";
    } else if (status == "Sangat Tidak Sehat") {
        return "Kondisi gas karbon dioksida tidak dapat
di kondisikan, segera mengamankan diri";
    } else (status == "Berbahaya") {
        return "Kondisi udara buruk, segera keluarkan
diri dari ruangan.";
    }
}

String getCO2Effect(String status) {
    if (status == "Aman") {
        return "Tidak ada efek negatif pada kesehatan.";
    } else if (status == "Sedang") {
        return "Mungkin menyebabkan sedikit
ketidaknyamanan pada beberapa orang.";
    }
}
```

```

} else if (status == "Tidak Sehat") {
    return "Kesehatan mulai terganggu";
} else if (status == "Sangat Tidak Sehat") {
    return "Gangguan pernapasan";
} else (status == "Berbahaya") {
    return "Paparasi berkepanjangan dapat menyebabkan
sakit kepala, pusing, dan koma, hingga kematian";
}
}

void handleRoot() {
    String htmlPage = getHTML(); // Mendapatkan
halaman HTML dari file "index.h"
    server.send(200, "text/html", htmlPage);
}

String getHTML() {
    return FPSTR(htmlPage); // Mengambil isi halaman
HTML dari file "index.h"
}

```

## Lampiran 2. Code System (HTML Arduino IDE)

```

const char* htmlPage = R"(
<!DOCTYPE html>
<html>
<head>
    <title>CO2 Gas Monitor</title>
    <style>
        body {
            text-align: center;
            font-family: Arial, sans-serif;
            background-image: linear-gradient(to
bottom, #f2f2f2, #e0e0e0, #ffffff);
            background-repeat: no-repeat;
            background-size: cover;
            background-attachment: fixed;
            margin: 0;
            padding: 0;
            color: #333;
        }
    </style>

```

```
h1 {
  margin-top: 20px;
}
.table-container {
  display: flex;
  justify-content: space-around;
  margin-top: 20px;
}
table {
  border-collapse: collapse;
  width: 40%;
  border: 2px solid #888;
  border-radius: 8px;
  box-shadow: 0 2px 4px rgba(0, 0, 0,
0.1);
}
th, td {
  border: 1px solid black;
  padding: 10px;
  text-align: center;
}
th {
  background-color: #4CAF50;
  color: white;
}
canvas {
  max-width: 80%;
  margin-top: 20px;
}
#clock {
  font-size: 24px;
  margin-top: 20px;
}
.digital-clock {
  display: inline-block;
  background-color: #333;
  color: #00bcd4;
  border-radius: 5px;
  padding: 10px;
  font-size: 24px;
}
```

```

        /* Tambahkan CSS untuk warna latar belakang
tabel */
        .table-container table {
            background-color: var(--table-bg-color);
        }
    </style>
    <script
src="https://cdn.jsdelivr.net/npm/chart.js"></script
>
</head>
<body>
    <h1>CO2 Gas Monitor</h1>
    <div class="digital-clock" id="clock"></div>

    <div class="table-container">
        <table>
            <tr>
                <th>Waktu</th>
                <th>Kadar CO2 (ppm)</th>
            </tr>
            <tr>
                <td id="timeCO2">-</td>
                <td id="co2Value">0</td>
            </tr>
        </table>

        <table>
            <tr>
                <th>Waktu</th>
                <th>Status</th>
                <th>Saran</th>
                <th>Efek Paparan</th>
            </tr>
            <tr>
                <td id="timeStatus">-</td>
                <td id="status">-</td>
                <td id="advice">-</td>
                <td id="effect">-</td>
            </tr>
        </table>
    </div>

```

```
<canvas id="chart"></canvas>

<script>
    var socket = new WebSocket('ws://' +
window.location.hostname + ':81/');
    var chartLabels = [];
    var chartData = [];
    var chartColors = [];
    var highestCO2Value = 0; // Simpan nilai CO2
    tertinggi

    var ctx =
document.getElementById('chart').getContext('2d');
    var chart = new Chart(ctx, {
        type: 'line',
        data: {
            labels: chartLabels,
            datasets: [{
                label: 'Kadar CO2 (ppm)',
                borderColor: chartColors,
                data: chartData,
                fill: false
            }]
        },
        options: {
            scales: {
                y: {
                    beginAtZero: true
                }
            },
            animation: {
                duration: 0
            }
        }
    });

    function updateTime() {
        var now = new Date();
        var timeCO2 = now.getHours() + ":" +
now.getMinutes() + ":" + now.getSeconds();
    }
}
```



```

        document.getElementById('timeCO2').inner
Text = timeCO2;
        document.getElementById('timeStatus').in
nerText = timeCO2;
        document.getElementById('clock').innerTe
xt = now.toLocaleTimeString();
    }

    function fetchData() {
        socket.send('fetchData');
    }

    socket.onmessage = function(event) {
        var data = JSON.parse(event.data);
        document.getElementById('co2Value').inne
rText = data.co2;
        document.getElementById('status').innerT
ext = data.status;
        document.getElementById('advice').innerT
ext = data.advice;
        document.getElementById('effect').innerT
ext = data.effect;

        updateTime();

        chartLabels.push(document.getElementById
('timeCO2').innerText);
        chartData.push(data.co2);
        chartColors.push(getColor(data.status));
// Ambil warna berdasarkan status
        if (chartLabels.length > 10) {
            chartLabels.shift();
            chartData.shift();
            chartColors.shift();
        }
        chart.update();

        // Update nilai CO2 tertinggi dan atur
warna latar belakang tabel
        highestCO2Value =
Math.max(highestCO2Value, data.co2);

```

```
        var tableContainer =
document.querySelector('.table-container');
        if (data.co2 === highestCO2Value) {
            tableContainer.style.setProperty('--
table-bg-color', 'red');
        } else {
            tableContainer.style.setProperty('--
table-bg-color', 'green');
        }
    };

    function getColor(status) {
        if (status === 'Aman') {
            return 'rgb(75, 192, 192)';
        } else if (status === 'Sedang') {
            return 'rgb(255, 165, 0)';
        } else {
            return 'rgb(255, 0, 0)';
        }
    }

    updateTime();
    fetchData(); // Mengambil data saat halaman
dimuat
        setInterval(fetchData, 5000); // Mengambil
data setiap 5 detik sekali
        setInterval(updateTime, 1000);
    </script>
</body>
</html>
)";
```