

ALAT PEMANTAU KUALITAS UDARA BERBASIS *IOT*

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan
Diploma III Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh:

Muhammad Abdurahman Alfarizi NIRM: 0032116

Tiara Yasmin Arimbi Chantika NIRM: 0032159

POLITEKNIK MANUFAKTUR NEGERI

BANGKA BELITUNG

TAHUN 2024

LEMBAR PENGESAHAN

ALAT PEMANTAU KUALITAS UDARA BERBASIS IOT

Oleh:

Muhammad Abdurahman Alfarizi/0032116

Tiara Yasmin Arimbi Chantika/0032159

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu kelulusan
Program Diploma III Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui,

Pembimbing 1



Eko Sulistyono, M.T.

Pembimbing 2



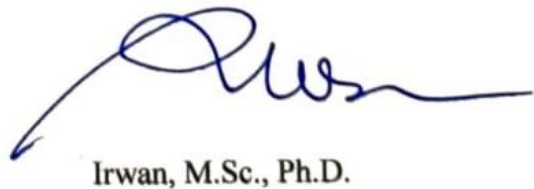
I Made Andik Setiawan, M.Eng., Ph.D

Penguji 1



Zanu Saputra, M.Tr.T.

Penguji 2



Irwan, M.Sc., Ph.D.

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini:

Nama Mahasiswa 1 : Muhammad Abdurahman Alfarizi NIRM: 0032116

Nama Mahasiswa 2 : Tiara Yasmin Arimbi Chantika NIRM: 0032159

Dengan Judul : Alat Pemantau Kualitas Udara Berbasis IoT

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungai Liat, 25 Juli 2024


Nama Mahasiswa

Tanda Tangan

1. Muhammad Abdurahman Alfarizi


.....

2. Tiara Yasmin Arimbi Chantika


.....

ABSTRAK

Beberapa penyebab utama penyakit pernapasan dan kardiovaskular disebabkan polusi udara buruk. Tidak semua wilayah memiliki SPKUA contohnya seperti Sungai Liat, Bangka Belitung. Wilayah dengan polusi udara parah menjadi prioritas pemasangan SPKUA. Selain itu, faktor biaya dan instalasi yang kompleks menjadi penghalang penyebaran SPKUA. Untuk mengatasi masalah tersebut, penelitian ini mengembangkan alat pemantau kualitas udara berbasis IoT yang lebih efisien, mudah digunakan, mudah dibawa dan berbasis IoT. Penelitian ini bertujuan untuk membuat alat pemantau kualitas udara berbasis IoT dengan menggunakan ESP32, sensor MQ dan sensor DHT22. Prosesnya meliputi perancangan, dan pembuatan program alat pemantau kualitas udara, perancangan electrical dan hardware lalu pembuatan dan penggabungan software dan hardware, serta pengujian pada alat pemantau kualitas udara. bahwa alat pemantau kualitas udara berbasis IoT ini dapat mendeteksi dan mengukur faktor-faktor penting untuk kualitas udara. Secara keseluruhan, tujuan proyek ini untuk menyediakan solusi pemantauan kualitas udara yang efisien dan efektif telah dicapai. Sistem ini dapat memberikan informasi kondisi udara secara real-time, yang sangat bermanfaat untuk tujuan lingkungan, kesehatan, dan keselamatan publik.

Kata kunci: Internet of Things (IoT), Kualitas Udara, Monitoring, Node-RED.

ABSTRACT

Some of the main causes of respiratory and cardiovascular diseases are due to bad air pollution. Not all areas have SPKUAs, such as Sungai Liat, Bangka Belitung. Areas with worse air pollution are prioritized for SPKUA installation. In addition, cost and complex installation factors are barriers to SPKUA distribution. To overcome these problems, this research developed an air quality monitoring device IoT-based that is more efficient, easy to use, portable and IoT-based. This research aims to create an IoT-based air quality monitoring device using ESP32, MQ sensor and DHT22 sensor. The process includes designing, and creating an air quality monitoring program, designing electrical and hardware then making and integrating software and hardware, and testing the air quality monitoring device. That this IoT-based air quality monitoring device can detect and measure important factors for air quality. Overall, the goal of this project to provide an efficient and effective air quality monitoring solution has been achieved. The system can provide real-time air condition information, which is highly beneficial for environmental, health, and public safety purposes.

Keywords: Air quality, Internet of Things (IoT), Monitoring, Node-RED.

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Allah SWT atas rahmat dan karunia-Nya penulis dapat menyusun dan menyelesaikan laporan Proyek Akhir (PA) dengan judul “Alat Pemantau Kualitas Udara Berbasis *IoT*” ini dengan baik dan tepat waktu.

Pembuatan laporan ini bertujuan sebagai salah satu syarat untuk memperoleh gelar Diploma III Politeknik Manufaktur Negeri Bangka Belitung. Penulis berharap pembaca dapat memahami proyek akhir dari penulis. Penyusunan laporan proyek akhir ini didasarkan pada pengembangan jurnal – jurnal penelitian sebelumnya. Penulis mencoba menerapkan ilmu yang telah penulis peroleh selama tiga tahun menempuh pendidikan di Politeknik Manufaktur Negeri Bangka Belitung.

Pada kesempatan ini penulis ingin menyampaikan rasa terima kasih kepada orang – orang yang telah berperan dalam membimbing dan membantu penulis untuk menyelesaikan laporan proyek akhir, seperti:

1. Keluarga penulis yang selalu memberikan dukungan baik secara moral maupun materi sehingga penulis dapat memberikan hasil yang terbaik.
2. Bapak I Made Andik Setiawan, M.Eng., Ph.D. selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung sekaligus dosen pembimbing 2 dalam proyek akhir ini.
3. Bapak Eko Sulistyono, M.T. selaku Wakil Direktur III Politeknik Manufaktur Negeri Bangka Belitung sekaligus dosen pembimbing 1 dalam proyek akhir ini.
4. Bapak Zanu Saputra, S.ST., M.Tr.T. selaku Ketua Jurusan Teknik Elektro dan Informatika Politeknik Manufaktur Negeri Bangka Belitung.
5. Bapak Ocsirendi, M.T. selaku Ketua Program Studi DIII Teknik Elektronika Politeknik Manufaktur Negeri Bangka Belitung.

6. Bapak Agus Dwi Yunto selaku alumni Politeknik Manufaktur Negeri Bangka Belitung yang telah banyak membantu penulis untuk menyelesaikan kendala yang dihadapi saat pengerjaan alat.
7. Seluruh tenaga pendidik dan kependidikan Politeknik Manufaktur Negeri Bangka Belitung.
8. Rekan – rekan mahasiswa Politeknik Manufaktur Negeri Bangka Belitung dan rekan kelompok proyek akhir ini yang telah membantu dalam penyelesaian proyek akhir.
9. Seluruh pihak yang terlibat baik secara langsung maupun tidak langsung dalam penyelesaian proyek akhir ini yang tidak bisa disebutkan satu persatu.

Penulis menyadari masih banyak kekurangan dalam penyusunan laporan ini karena keterbatasan pengetahuan dan keterampilan penulis. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun dari semua pihak yang terlibat untuk memperbaiki dan mengembangkan lebih lanjut makalah ini di masa mendatang. Penulis berharap makalah ini dapat bermanfaat khususnya bagi yang berminat terhadap perkembangan ilmu pengetahuan dan teknologi.

Sungai Liat, 25 Juli 2024

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN	ii
PERNYATAAN BUKAN PLAGIAT	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan Proyek Akhir	2
1.4. Batasan Masalah.....	2
BAB II DASAR TEORI.....	3
2.1 Kualitas Udara.....	3
2.2. Indeks Standar Pencemaran Udara (ISPU)	4
2.3 Teknologi <i>Internet of Things (IoT)</i>	5
2.4 Mikrokontroler ESP32	6
2.5 Sensor Pendeteksi Kadar Gas, Kelembaban dan Suhu.....	7
2.5.1 Sensor MQ-2.....	7
2.5.2 Sensor MQ-8.....	8
2.5.3 Sensor MQ-9.....	9
2.5.4 Sensor MQ-135.....	10
2.5.5 Sensor DHT22	11
2.6 Set Top <i>Box</i>	12
2.7 Komunikasi <i>Modbus</i>	13
2.8 <i>Node-Red</i>	13

BAB III METODE PELAKSANAAN.....	15
3.1 Studi Literatur.....	16
3.2 Perancangan dan Perakitan <i>Prototype</i> Alat Pemantau Kualitas Udara	17
3.3 Pembuatan Program Alat Pemantau Kualitas Udara	17
3.4 Perancangan <i>Electrical</i> dan <i>Hardware</i> Alat Pemantau Kualitas Udara	18
3.4.1 Perancangan <i>Box</i> Alat.....	18
3.4.2 Perancangan Elektrik	19
3.5 Pembuatan dan Penggabungan Program dan <i>Hardware</i> Alat Pemantau Kualitas Udara	20
3.6 Pengujian Alat Pemantau Kualitas Udara	21
3.7 Pembuatan Laporan Proyek Akhir	21
BAB IV PEMBAHASAN.....	22
4.1 Deskripsi Alat Pemantau Kualitas Udara Berbasis <i>IoT</i>	22
4.2 Diagram Blok Alat Pemantau Kualitas Udara Berbasis <i>IoT</i>	23
4.3 Prinsip Kerja Alat Pemantau Kualitas Udara Berbasis <i>IoT</i>	24
4.4 Pembuatan Program Alat Pemantau Kualitas Udara	24
4.4.1 Pembuatan Program Mikrokontroler Alat Pemantau Kualitas Udara... 24	
4.5 Pembuatan dan Penggabungan Program dan <i>Hardware</i> Alat Pemantau Kualitas Udara	25
4.5.1 Pembuatan <i>Hardware</i> Mekanik	25
4.5.2 Pembuatan <i>Hardware</i> Elektrik	25
4.5.3 Penggabungan Program dan <i>Hardware</i>	25
4.6 Pengujian Sensor	26
4.6.1 Pengujian Sensor MQ-2.....	26
4.6.2 Pengujian Sensor MQ-8.....	28
4.6.3 Pengujian Sensor MQ-9.....	30
4.6.4 Pengujian Sensor MQ-135.....	32
4.6.5 Pengujian Sensor DHT22	34
4.7 Pengujian <i>Node-Red</i>	35
4.7.1 <i>Flow Node-Red</i>	36
4.8 Pengujian Keseluruhan Alat Pemantau Kualitas Udara Berbasis <i>IoT</i>	36

4.8.1 Hasil Pengujian Keseluruhan Sistem dan Alat	36
BAB V PENUTUP.....	40
5.1 Kesimpulan.....	40
5.2 Saran.....	40
DAFTAR PUSTAKA.....	41
LAMPIRAN.....	45



DAFTAR TABEL

Tabel 2.1 Kategori ISPU	5
Tabel 4.1 Hasil Pengujian Sensor MQ-2.....	28
Tabel 4.2 Hasil Pengujian Sensor MQ-8.....	29
Tabel 4.3 Hasil Pengujian Sensor MQ-9.....	31
Tabel 4.4 Hasil Pengujian Sensor MQ-135.....	33
Tabel 4.5 Hasil Pengujian Sensor DHT22	35
Tabel 4.6 Hasil Pengujian Keseluruhan Sistem dan Alat.....	38
Tabel 4.7 Nilai Hasil <i>Air Quality Detector</i>	38



DAFTAR GAMBAR

Gambar 2.1 Mikrokontroler ESP 32	7
Gambar 2.2 Sensor MQ-2	8
Gambar 2.3 Sensor MQ-8	9
Gambar 2.4 Sensor MQ-9	10
Gambar 2.5 Sensor MQ-135	11
Gambar 2.6 Sensor DHT 22	12
Gambar 3.1 Flowchart Metode Pelaksanaan	16
Gambar 3.2 Tampak Atas Desain Box	18
Gambar 3.3 Tampak Samping Desain Box	19
Gambar 3.4 Tampak Keseluruhan Box	19
Gambar 3.5 Skematik Rangkaian Alat Pemantau Kualitas Udara	20
Gambar 4.1 Diagram Blok Alat Pemantau Kualitas Udara Berbasis IoT	23
Gambar 4.2 Hasil Alat Pemantau Kualitas Udara	26
Gambar 4.3 Diagram Blok Pengujian Sensor	26
Gambar 4.4 Skematik Rangkaian Pengujian Sensor MQ-2	27
Gambar 4.5 Skematik Rangkaian Pengujian Sensor MQ-8	29
Gambar 4.6 Skematik Rangkaian Pengujian Sensor MQ-9	30
Gambar 4.7 Skematik Rangkaian Pengujian Sensor MQ-135	32
Gambar 4.8 Skematik Rangkaian Pengujian Sensor DHT22	34
Gambar 4.9 Flow Node-Red	36
Gambar 4.10 Tampilan Website	37
Gambar 4.11 Tampilan Layar TV	37

DAFTAR LAMPIRAN

LAMPIRAN 1: Daftar Riwayat Hidup (Perorangan)

LAMPIRAN 2: Program Mikrokontroler

LAMPIRAN 3: *Flow Node-RED*



BAB I

PENDAHULUAN

1.1. Latar Belakang

Kualitas udara yang tidak memadai di dalam ruangan memiliki implikasi serius terhadap kesejahteraan penghuninya. Selain dapat menyebabkan masalah kesehatan yang mengganggu, kondisi udara dalam ruangan juga mempengaruhi kenyamanan dan tingkat produktivitas individu. Penting untuk memperhatikan kualitas udara karena dampak negatifnya dapat menyebabkan masalah kesehatan yang mengganggu [1]. Perubahan zaman juga telah mengubah pola perilaku manusia, mengakibatkan banyak individu menghabiskan sebagian besar waktu mereka di dalam ruangan, entah itu di rumah pribadi mereka, di tempat kerja profesional mereka, atau bahkan di fasilitas publik yang sering digunakan.

Berdasarkan data Kementerian Lingkungan Hidup dan Kehutanan pada 2023, Stasiun Pemantau Kualitas Udara Ambien (SPKUA) sebanyak 68 unit telah dibangun menyebar di berbagai wilayah di Indonesia [2]. Namun, tidak semua wilayah memiliki SPKUA seperti Sungai Liat, Bangka Belitung. Wilayah yang mengalami polusi udara yang parah menjadi prioritas utama pemasangan SPKUA. Selain itu, faktor biaya dan instalasi yang kompleks juga menjadi penghalang pemerataan penyebaran SPKUA [3].

Untuk mengatasi masalah yang dijelaskan sebelumnya, penelitian ini berkonsentrasi pada pengembangan alat pemantau kualitas udara berbasis *IoT* yang lebih efisien, mudah digunakan, mudah dibawa dan berbasis *IoT*. Alat ini diharapkan dapat memberikan data yang komprehensif dan akurat dengan menggabungkan sensor yang dapat mendeteksi berbagai polutan. Penggunaan *IoT* meningkatkan efektivitas pemantauan kualitas udara melalui sensor yang terhubung ke internet dan dapat mengirim data secara *real-time* sehingga data dapat dipantau melalui *smartphone* maupun layar monitor yang telah disiapkan.

Indeks Standar Pencemar Udara (ISPU) adalah nilai yang tidak bersifat satuan yang mencerminkan kualitas udara di suatu lokasi, berdasarkan efeknya terhadap

kesehatan manusia, keindahan dan kehidupan lainnya [4]. ISPU akan menjadi parameter utama dalam menetapkan standar kualitas udara pada alat yang sedang dikembangkan. Melalui Alat Pemantau Kualitas Udara Berbasis *IoT* yang dilengkapi dengan parameter yang akan dipantau berupa data sensor, kelembapan serta suhu udara.

1.2. Rumusan Masalah

Rumusan masalah untuk proyek akhir ini mencakup:

1. Bagaimana merancang dan membuat alat pemantau kualitas udara berbasis *IoT*?
2. Bagaimana membuat alat pemantau kualitas udara dengan parameter yang sesuai dengan rentang kategori Indeks Standar Pencemar Udara (ISPU)?
3. Bagaimana cara mendapatkan nilai sensor yang dapat ditampilkan secara *real-time*?

1.3. Tujuan Proyek Akhir

Berikut ini adalah tujuan proyek akhir ini:

1. Merancang dan membuat alat pemantau kualitas udara berbasis *IoT*.
2. Membuat sistem dan alat pemantau kualitas udara dengan parameter yang sesuai dengan rentang kategori Indeks Standar Pencemaran Udara (ISPU).
3. Memperoleh data sensor yang dapat ditampilkan ke dalam *website* mengenai kualitas udara secara *real-time*.

1.4. Batasan Masalah

Adapun batasan masalah dalam pengerjaan proyek akhir ini, seperti:

1. Rentang deteksi data maksimal yang dapat dihasilkan terbatas pada jarak 10 meter.
2. Sensor hanya mampu menyajikan informasi secara *real-time* karena tidak ada kapasitas untuk menyimpan data.
3. Sistem pemantau pada *website* hanya dapat diakses ketika terhubung dengan jaringan internet yang sama.

BAB II DASAR TEORI

2.1 Kualitas Udara

Kualitas udara merupakan alat untuk mengukur kondisi udara yang layak di sebuah area. Disebabkan oleh aktivitas manusia, udara menjadi lebih tercemar. Penurunan kualitas udara disebabkan oleh beberapa jenis gas, asap kendaraan, asap industri dan limbah rumah tangga [5]. Sistem pernapasan, kesehatan umum, dan produktivitas dapat terpengaruh oleh pencemaran udara. Hal ini disebabkan oleh zat-zat yang tercemar dalam udara yang masuk ke paru-paru dan menyebabkan peradangan. Karena kemampuan sistem kekebalan tubuh yang lebih lemah, anak-anak lebih rentan terhadap peradangan ini daripada orang dewasa [6].

Alat pemantau kualitas udara berbasis *IoT* ini menggunakan sensor MQ untuk mendeteksi kadar gas serta sensor DHT 22 untuk mengukur kelembaban dan suhu. Sangat penting untuk memantau kadar gas-gas ini di udara untuk memastikan kesehatan dan keselamatan manusia. Adapun kadar aman untuk beberapa faktor yang akan diukur dalam alat ini, seperti:

1. LPG (*Liquefied Petroleum Gas*)

Ambang batas untuk paparan gas LPG dalam jangka pendek adalah 1000 ppm [7].

2. *Methane* (CH_4)

Karena metana tidak beracun, tidak ada batas paparan yang ditetapkan untuk metana. Namun, kadar yang aman untuk menghindari bahaya kebakaran adalah di bawah 5% volume udara (50,000 ppm) [8]. Pada ruangan tertutup, kadar aman CH_4 kurang dari 1% volume di udara (10,000 ppm)

3. *Hidrogen* (H_2)

Meskipun sangat mudah terbakar, hidrogen tidak beracun. Pedoman OSHA menekankan bahwa untuk mencegah ledakan, ventilasi harus menjaga konsentrasi hidrogen di bawah 1% untuk mengurangi risiko ledakan [9]. Ada pula pengendalian hidrogen yang aman yaitu, tetap menjaga konsentrasi udara

tetap di bawah *Lower Explosive Limit* (LEL) sebesar 4%. Pada ruangan tertutup, kadar aman CH₄ kurang dari 0.4% volume di udara (4,000 ppm).

4. *Karbon Monoksida (CO)*

Untuk waktu kerja 8 jam sehari, *Environmental Protection Agency* (EPA) menerapkan standar batas paparan karbon monoksida maksimal 9 ppm (*parts per million*) [10].

5. *Karbon Dioksida (CO₂)*

OSHA menyarankan batas aman CO₂ sebesar 5000 ppm untuk paparan panjang selama 8 jam [11].

6. *Kelembaban*

Tingkat kelembapan ideal berkisar antara 45% - 65% untuk menciptakan lingkungan sehat dan nyaman [12].

7. *Suhu*

Rentang suhu ideal secara umum agar tetap terasa nyaman dan aman yaitu, dalam suhu 20-24°C (68-76°F) [13].

2.2. Indeks Standar Pencemaran Udara (ISPU)

Menurut Keputusan Menteri Negara Lingkungan Hidup Nomor KEP 45 / MENLH / 1997 Tentang Indeks Standar Pencemar Udara, Indeks Standar Pencemar Udara (ISPU) adalah indeks standar kualitas udara yang digunakan secara resmi di Indonesia. Indeks Standar Pencemar Udara (ISPU) adalah nilai tanpa satuan yang mencerminkan kualitas udara di sebuah lokasi, berdasarkan efeknya pada kondisi kesehatan manusia, keindahan serta kehidupan lainnya [14].

Mengubah Keputusan Menteri Lingkungan Hidup nomor 45 tahun 1997 tentang Perhitungan, Pelaporan, dan Informasi Indeks Standar Pencemar Udara menjadi Peraturan Menteri Lingkungan Hidup dan Kehutanan nomor 14 tahun 2020 tentang Indeks Standar Pencemar Udara, yang dikeluarkan oleh KLHK pada tahun 2020. Pada peraturan pengganti ini, perhitungan ISPU dilakukan pada 7 (tujuh) parameter, yaitu PM₁₀, PM_{2.5}, NO₂, SO₂, CO, O₃, dan HC. Dua parameter, HC dan PM_{2.5}, ditambahkan dari peraturan sebelumnya karena besarnya risiko HC dan

PM2.5 bagi kesehatan manusia [15]. Berikut merupakan kategori Indeks Standar Pencemaran Udara:

Tabel 2.1 Kategori ISPU

Rentang	Kategori	Penjelasan
1 – 50	Baik	Tingkat mutu udara yang sangat baik, tidak memberikan efek negatif terhadap manusia, hewan dan tumbuhan.
51 – 100	Sedang	Tingkat mutu udara masih dapat diterima pada kesehatan manusia, hewan dan tumbuhan.
101 – 200	Tidak Sehat	Tingkat mutu udara yang bersifat merugikan pada manusia, hewan dan tumbuhan
201 – 300	Sangat Tidak Sehat	Tingkat mutu udara yang dapat meningkatkan risiko kesehatan pada sejumlah segmen populasi yang terpapar.
301+	Berbahaya	Tingkat mutu udara yang dapat merugikan kesehatan serius pada populasi dan perlu penanganan cepat.

2.3 Teknologi *Internet of Things (IoT)*

Internet of Things (IoT) adalah suatu gagasan untuk membantu berbagai perangkat fisik, sensor dan objek sehari-hari yang terkoneksi dengan internet untuk berkomunikasi antara satu sama lain dan dengan sistem komputer yang terpusat tanpa interaksi manusia secara langsung [16]. *IoT* juga memungkinkan perangkat-perangkat ini untuk mengumpulkan, berbagi dan memproses data.

Tiga komponen utama yang membentuk *Internet of Things (IoT)* pada Alat Pemantau Kualitas Udara Berbasis *IoT*, seperti:

- Sensor atau perangkat yang mengumpulkan data. Alat pemantau kualitas udara ini menggunakan 6 buah sensor, seperti: sensor MQ2, sensor MQ8, sensor MQ9, sensor MQ135 dan sensor DHT22.

- Konektivitas, dalam alat pemantau kualitas udara ini, komunikasi *Modbus* digunakan untuk menghubungkan mikrokontroler ESP32 dengan *set-top box*. Selain itu, alat ini terhubung ke jaringan internet, yang memungkinkan *platform* berbasis *IoT* untuk memantau data secara *real-time*.
- *Platform* atau sistem pengolahan data. Untuk memantau dan mengelola data kualitas udara secara *real-time*, alat ini menggunakan *platform IoT*. Mikrokontroler ESP32 mengumpulkan dan memproses data dari sensor – sensor yang digunakan. Kemudian, data dikirim ke *set-top box* melalui konektivitas *Modbus* dan diunggah ke *server* yang mendukung menampilkan dan analisis data. Melalui pengaplikasian *website* atau layar monitor, *platform* ini memungkinkan pengguna memantau kualitas udara secara efektif dan akurat.

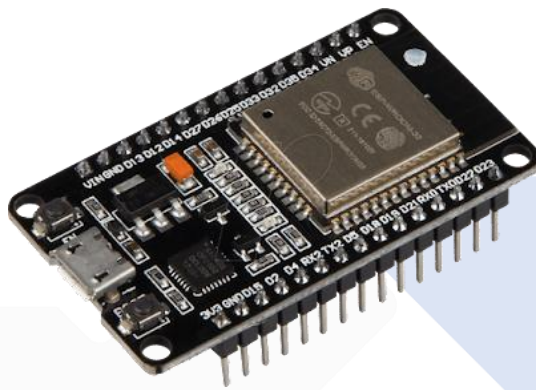
2.4 Mikrokontroler ESP32

ESP32 merupakan mikrokontroler yang telah terintegrasi dengan jaringan *Wifi* dan koneksi *Bluetooth*, yang dapat menyederhanakan dalam membuat sistem *IoT* yang membutuhkan akses nirkabel. Mikrokontroler ESP32 membaca *input* data dari sensor dan mengirimkannya melalui modul *WiFi* ESP32 ke server atau *platform IoT* untuk diproses. Setelah itu, data yang diterima ditampilkan ke halaman *platform*, sehingga pengguna dapat melihatnya [17].

Mikrokontroler ESP32 memiliki prosesor *dual-core*, kecepatan hingga 240 MHz, RAM 520 KB, dan memori *flash* hingga 4 MB. Sesuai untuk aplikasi *IoT* karena memiliki tegangan operasi 2.2V - 3.6V dan mengonsumsi sedikit daya. Dengan spesifikasi ini ESP32 digunakan untuk mengumpulkan data dari sejumlah sensor, mengolahnya, dan kemudian mengirimkannya ke server melalui koneksi internet. ESP32 mendukung protokol *Modbus* melalui UART, yang memungkinkan integrasi dengan *set-top box* untuk menampilkan data sensor.

Dalam proyek ini, fitur *Analog-to-Digital Converter (ADC)* pada ESP32 digunakan untuk membaca tegangan analog dari beberapa sensor kualitas udara, termasuk tetapi tidak terbatas pada MQ-2, MQ-8, MQ-9, dan MQ-135. ADC mengonversi tegangan analog sensor menjadi nilai digital 12-bit, yang

memungkinkan mikrokontroler untuk mengolah data ini dan menghitung konsentrasi gas dalam satuan ppm (*parts per million*). Selanjutnya, rumus kalibrasi khusus untuk setiap sensor digunakan untuk memproses data digital dan ditampilkan secara *real-time* pada *platform IoT* seperti *Node-RED*. Implementasi ini memastikan pemantauan kualitas udara yang akurat dan efisien, yang memungkinkan pengguna untuk mendapatkan informasi lingkungan secara cepat dan tepat.



Gambar 2.1 Mikrokontroler ESP32

2.5 Sensor Pendeteksi Kadar Gas, Kelembaban dan Suhu

2.5.1 Sensor MQ-2

Sensor MQ-2 merupakan suatu perangkat sensor gas yang mampu mendeteksi LPG, Butana, Metana dan asap. Umumnya, sensor ini dimanfaatkan untuk mengidentifikasi kebocoran gas LPG, serta mendeteksi adanya perokok. Sensor ini mengandung material sensitif yang mengalami perubahan resistansi saat berinteraksi dengan gas-gas tersebut. Variasi nilai resistansi ini kemudian dimanfaatkan untuk mendeteksi keberadaan gas-gas tersebut [18].

Sensor gas MQ-2 memiliki rentang konsentrasi 300 hingga 10.000 ppm dan beroperasi pada tegangan 5V DC dengan resistansi 5k Ω . Sensor ini memiliki waktu respons cepat 10 detik dan untuk pengoperasian yang stabil membutuhkan waktu respons 20-30 detik. Sensor ini dapat berfungsi pada suhu dari -20°C hingga 50°C. Untuk keluarannya sensor ini menggunakan *output* analog yang dapat dibaca oleh mikrokontroler. Pengujian yang dilakukan (Sohibun, Daruwati, Hatika, &

Mardiansyah, 2021) bahwa kesalahan pembacaan dari sensor MQ-2 pada berbagai kondisi ruangan tidak melebihi 10% [19]. Sensor ini mempunyai 4 pin, tetapi hanya 3 pin yang digunakan, yaitu:

- Pin 1: VCC
- Pin 2: GND
- Pin 3: Ao, pin ini terhubung ke pin 34 mikrokontroler ESP32



Gambar 2.2 Sensor MQ-2

2.5.2 Sensor MQ-8

Sensor MQ-8 sensitif terhadap gas hidrogen, sering kali sensor ini sering digunakan untuk mendeteksi konsentrasi hidrogen dalam atmosfer dengan tujuan pencegahan kebakaran [20]. Perubahan konsentrasi gas target memengaruhi resistensi komponen sensor maka dari itu *output* yang dihasilkan pun akan berbeda ketika konsentrasi gas terdeteksi. Penelitian yang dilakukan (Sunny, Zhao, Li, & Sakiliba, 2021) menunjukkan bahwa sensor MQ-8 dapat mengidentifikasi peningkatan konsentrasi hidrogen bahkan ketika parameter lingkungan lainnya seperti suhu dan kelembapan berubah [21].

Sensor gas MQ-8 mengukur konsentrasi gas dengan cepat dan akurat pada 100 – 10.000 ppm dengan waktu respons yang cepat dan elektroda internal dengan sensitivitas tinggi. Ini memungkinkan sensor untuk mencapai kondisi kerja yang ideal. Sensor ini terdiri dari bahan tin dioksida (SnO_2) yang resistensinya berubah saat terpapar gas hidrogen. Sensor ini mengeluarkan *output* analog yang dapat dibaca oleh mikrokontroler dan berfungsi pada tegangan 5V DC. Sensor ini mempunyai 4 pin, tetapi hanya 3 pin yang digunakan, yaitu:

- Pin 1: VCC
- Pin 2: GND
- Pin 3: Ao, pin ini terhubung ke pin 32 mikrokontroler ESP32



Gambar 2.3 Sensor MQ-8

2.5.3 Sensor MQ-9

Sensor MQ-9 dapat mengidentifikasi tingkat kebocoran dari silinder gas dan paling peka terhadap kandungan gas karbon monoksida. Sensor ini juga dapat mendeteksi gas seperti metana, LPG, CO, dan lainnya yang mudah terbakar [22]. Sensor ini dapat mendeteksi kepadatan karbon monoksida dari 10 ppm hingga 1000 ppm dan kepadatan gas yang mudah terbakar dari 100 ppm hingga 10.000 ppm. Pemanas internal MQ 9 akan memanaskan ketika tegangan 5V diterapkan. Hambatan internal sensor ini mengalami perubahan sesuai dengan berubahnya kerapatan dari gas yang dideteksi. Pengaplikasian yang dilakukan oleh (Nari, Laksono, Hidayat, & Udin, 2021) menyatakan bahwa nilai eror dari hasil data sensor yang dilakukan pada 6 pengujian adalah 8% [23]. Sensor ini mempunyai 4 pin, tetapi hanya 3 pin yang digunakan, yaitu:

- Pin 1: VCC
- Pin 2: GND
- Pin 3: Ao, pin ini terhubung ke pin 35 mikrokontroler ESP32



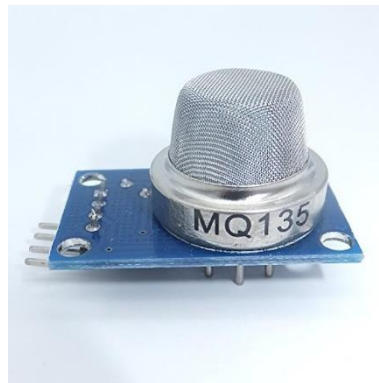
Gambar 2.4 Sensor MQ-9

2.5.4 Sensor MQ-135

Sensor MQ-135 dapat mendeteksi berbagai jenis gas beracun, seperti Amonia (NH_3), Benzena (C_6H_6), dan Karbon dioksida (CO_2). Saat mendeteksi berbagai jenis gas beracun, sensor ini memiliki respons yang stabil dan sesuai dengan ekspektasi [24]. Respons yang ditunjukkan oleh MQ-135 adalah bahwa jika kadar gas di ruangan lebih tinggi maka sensor akan mendeteksi kontaminasi udara oleh gas dengan lebih cepat. Ini karena, dalam kondisi normal sensor memiliki tahanan pada sensor yang cukup tinggi, tetapi ketika sensor mendeteksi kontaminasi udara oleh gas, tahanan pada sensor semakin menurun. Akibatnya, tegangan *output* keluaran sensor pun akan mengalami perbedaan apabila sensor berada dalam keadaan netral maupun apabila sensor mendeteksi kontaminasi udara oleh gas.

Hasil implementasi sensor MQ-135 pada penelitian *Smart Air Quality Guardian* menunjukkan bahwa, dengan rata-rata error sebesar 1,3%, sensor ini dapat diandalkan untuk mendeteksi tingkat gas CO_2 yang cukup akurat [25]. Dengan rentang deteksi 10-300 ppm amonia, 10-1000 ppm benzena, dan 10-300 ppm alkohol, MQ-135 sangat sensitif. Sensor ini membutuhkan tegangan operasi 5V dan memiliki elemen pemanas internal untuk memastikan kondisi operasi yang ideal. *Output* sensor menghasilkan tegangan analog yang proporsional dengan konsentrasi gas yang terdeteksi, yang memungkinkan mikrokontroler untuk membacanya dengan mudah. Sensor ini mempunyai 4 pin, tetapi hanya 3 pin yang digunakan, yaitu:

- Pin 1: VCC
- Pin 2: GND
- Pin 3: Ao, pin ini terhubung ke pin 33 mikrokontroler ESP32



Gambar 2.5 Sensor MQ-135

2.5.5 Sensor DHT22

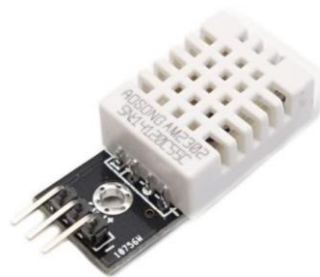
Sensor DHT terdiri atas termistor tipe NTC (*Negative Temperature Coefficient*) yang berfungsi sebagai pengukur suhu sekaligus kelembapan, dan juga merupakan sensor kelembapan yang tahan menghadapi risiko perubahan kandungan uap air di udara [26]. Sensor DHT22 sangat akurat dan dapat mengukur suhu dengan kesalahan $\pm 0,5$ derajat Celsius dan kelembapan relatif dengan kesalahan 2–5%. Suhu dapat berkisar antara -40 derajat Celsius dan 80 derajat Celsius [27]. Pada pengaplikasian yang dilakukan oleh (Sihombing & Listiari, 2020) menunjukkan bahwa kesalahan pembacaan sensor tidak lebih dari 5%, yang mengkonfirmasi bahwa sensor memberikan hasil yang akurat dan dapat diuji pada perangkat lainnya [28]. Saat pengukuran aktif, sensor ini memiliki arus operasional sekitar 2,5 mA dan dalam mode *idle* 40-50 μA . Sensor ini beroperasi pada tegangan 3.3V hingga 6V.

Keluaran dari sensor DHT22 merupakan data dari sensor berbentuk digital melalui pin 2. Sensor DHT 22 mengirimkan data berukuran 40bit dan terdiri dari 5 *byte*. *Byte* 1 – 2 untuk kelembapan dan pada kelembapan dalam persen dikodekan dalam 16 bit. Untuk *byte* 3 – 4 digunakan untuk suhu dan dikodekan dalam 16 bit. Dalam perhitungan suhu, jika bit ke-16 adalah 1, maka suhunya negatif dan

sebaliknya jika bit ke-16 adalah 0, maka suhunya positif. Hasil data dari kelembaban dan suhu berupa biner lalu dikonversikan ke desimal. Setelah dikonversikan ke desimal, data tersebut dimasukkan ke dalam rumus di bawah untuk mendapatkan *output* kelembaban dan suhu yang dapat dibaca dalam layar monitoring.

Sensor DHT 22 memiliki 3 pin yang dihubungkan, yaitu:

- Pin 1: VCC
- Pin 2: Out, pin ini terhubung ke pin 14 mikrokontroler ESP32
- Pin 3: GND



Gambar 2.6 Sensor DHT 22

2.6 Set Top Box

Set-top box (STB) ialah suatu perangkat elektronik untuk menerima dan mengonversi gelombang digital ke dalam format yang dapat disajikan di layar televisi atau perangkat tampilan lainnya. Sinyal tersebut dapat berasal dari berbagai sumber, seperti televisi kabel, satelit, atau layanan *streaming* internet. STB dapat digunakan sebagai hub untuk mengumpulkan, memproses, dan menampilkan data dari sensor yang terhubung dalam proyek *Internet of Things (IoT)* [29].

Pada alat pemantau kualitas udara ini, *set-top box (STB)* berfungsi sebagai tempat untuk mengolah dan menampilkan data dari berbagai sensor yang terhubung melalui mikrokontroler ESP32. STB menerima data dari sensor, seperti MQ-2, MQ-8, MQ-9, MQ-135 dan DHT22, kemudian data diproses dan ditampilkan pada layar atau monitor. Dengan menggunakan STB, antarmuka yang mudah diakses memungkinkan pengguna melihat kualitas udara secara langsung dan akurat. STB

juga dapat memberikan peringatan jika parameter kualitas udara berada di luar batas aman.

2.7 Komunikasi *Modbus*

Komunikasi *Modbus* adalah protokol serial komunikasi untuk menghubungkan berbagai peralatan elektronik, khususnya di lingkungan industri.. Protokol ini menggunakan konektor serial seperti RS-232 maupun RS-485 yang memungkinkan pertukaran data di antara perangkat. Satu perangkat berfungsi sebagai *master*, yang mengontrol komunikasi, dan satu atau lebih perangkat berfungsi sebagai *slave*, yang menerima perintah dari master dan menyelesaikan tugasnya [30].

Alat pemantau kualitas udara berbasis *IoT* ini menggunakan komunikasi *Modbus* untuk menghubungkan mikrokontroler ESP32 dengan *set top box*. Protokol *Modbus* memungkinkan mikrokontroler bertindak sebagai *master*, mengirimkan perintah, dan menerima data dari *set top box* sebagai *slave*. Dengan ini memungkinkan pengiriman data antara mikrokontroler dan STB untuk tujuan pengawasan dan kontrol. Data dari sensor dapat ditampilkan secara *real-time* di STB karena penggunaan *Modbus*, yang memungkinkan transfer data yang stabil dan andal.

2.8 *Node-Red*

Node-RED adalah alat pengembangan berbasis aliran yang awalnya dikembangkan oleh IBM untuk menghubungkan perangkat keras, API dan layanan *online* dengan cara baru. Alat ini memungkinkan pengguna merancang aplikasi dengan menggunakan editor berbasis *browser* yang memungkinkan pengaturan aliran data dengan *drag-and-drop*. *Node-RED* menggunakan alur yang berisi simpul-simpul (*nodes*) untuk merancang logika aplikasi dengan cara yang mudah dipahami dan visual, sehingga memudahkan pengembangan aplikasi [31].

Node-RED dapat berfungsi pada berbagai *platform*, mulai dari komputer desktop hingga perangkat tertanam seperti *Raspberry Pi*. Selain itu, mendukung berbagai protokol komunikasi seperti HTTP, MQTT, dan *WebSockets* dan memiliki

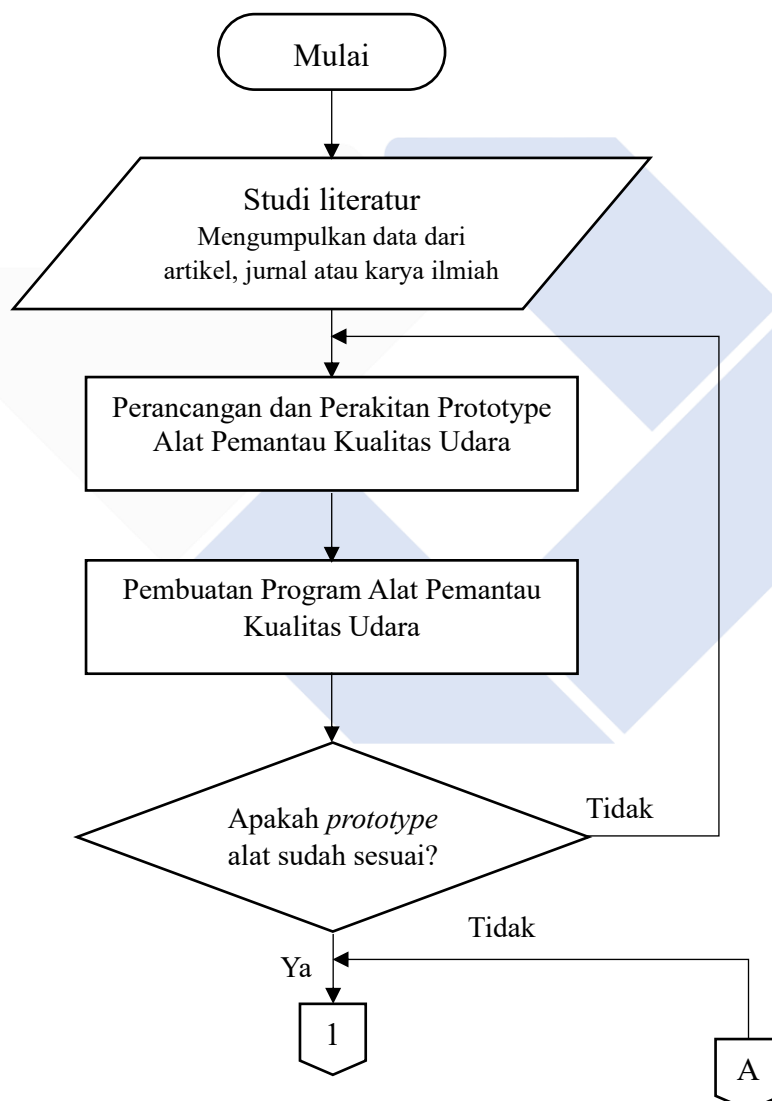
library yang kaya akan *node* yang dapat diintegrasikan dengan berbagai layanan dan perangkat, *Node-RED* sangat fleksibel dalam memenuhi berbagai kebutuhan pengembangan aplikasi dan sistem [32].

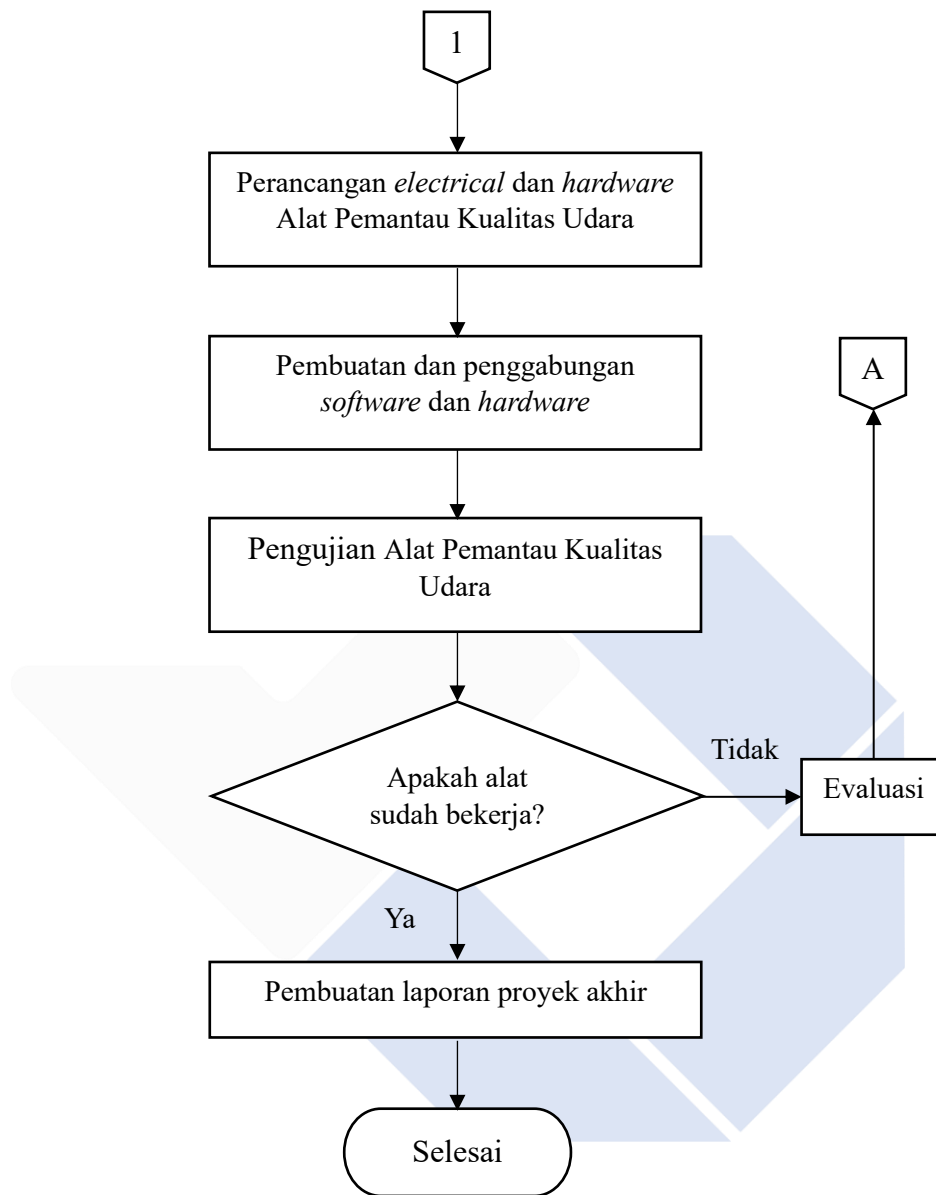
Pada alat pemantau kualitas udara ini, *Node-RED* digunakan untuk memproses dan menampilkan data dari sensor – sensor yang digunakan. Dengan menggunakan *Node-RED*, data yang dikumpulkan dari berbagai sensor yang terhubung ke mikrokontroler ESP32 dapat diproses dan divisualisasikan secara *real-time* melalui komunikasi *Modbus* ke dalam *server Node-RED*.



BAB III METODE PELAKSANAAN

Pada pelaksanaan proyek akhir dengan judul Alat Pemantau Kualitas Udara Berbasis *IoT* ini, memiliki beberapa tahapan dalam proses pengerjaannya. Di bawah ini merupakan diagram alir atau *flowchart* tahapan proses pengerjaan proyek akhir pada Gambar 3.1.





Gambar 3.1 *Flowchart* Metode Pelaksanaan

3.1 Studi Literatur

Ketika memulai proyek akhir ini, dilakukan studi literatur yang bertujuan untuk mengembangkan alat pemantau kualitas udara berbasis *IoT* ini. Fokus utama dari tahap ini adalah untuk mendapatkan pemahaman menyeluruh mengenai konsep dan teknologi yang akan digunakan, mencari penelitian-penelitian terdahulu yang relevan, serta melakukan pengumpulan data dan berbagai informasi yang

dibutuhkan sebagai dasar pengembangan sistem. Studi literatur ini mempelajari konsep dasar yang relevan dengan proyek ini, seperti kualitas udara, Indeks Standar Pencemar Udara (ISPU), teknologi *Internet of Things (IoT)*, dan spesifikasi teknis mengenai perangkat keras yang hendak diaplikasikan, seperti mikrokontroler ESP32, sensor MQ-2, MQ-8, MQ-9, MQ-135 dan DHT22.

3.2 Perancangan dan Perakitan *Prototype* Alat Pemantau Kualitas Udara

Pada tahap ini, perancangan dan perakitan *prototype* alat pemantau kualitas udara dilakukan secara sistematis untuk memastikan fungsionalitas dan akurasi alat yang dibuat. Langkah pertama melibatkan pemilihan komponen utama, yaitu sensor kualitas udara dan mikrokontroler ESP-32.

Setelah rancangan dibuat, selanjutnya *hardware* dirakit. Setelah itu, program *Arduino* diunggah ke mikrokontroler ESP-32, yang memungkinkan untuk mengontrol sensor dan memproses data yang diperoleh. Fungsi *Modbus* ditambahkan ke dalam program *Arduino* untuk memungkinkan komunikasi antara ESP-32 dan *platform Node-Red*, yang berfungsi sebagai *server* untuk menampilkan data kualitas udara secara *real-time*. Setelah itu, *prototype* yang telah dirakit diuji untuk memastikan bahwa semua komponen berfungsi dengan baik dan bahwa data yang dihasilkan akurat.

3.3 Pembuatan Program Alat Pemantau Kualitas Udara

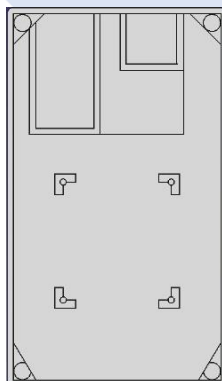
Pada tahap ini, program untuk alat pemantau kualitas udara dirancang dan diimplementasikan. Dimulai dengan menulis kode pada *platform Arduino IDE* yang kemudian diunggah ke mikrokontroler ESP-32. Program ini mencakup pengaturan parameter untuk sensor kualitas udara seperti sensor gas suhu dan kelembapan, serta integrasi fungsi *Modbus* untuk komunikasi data. Setelah program selesai ditulis, perangkat diuji di lingkungan nyata untuk memastikan keandalan dan akurasi keandalan alat dalam memantau kualitas udara. Sensor akan mengirimkan data ke *server Node-Red* untuk diproses lebih lanjut dan hasilnya akan ditampilkan melalui antarmuka *web* yang mudah diakses oleh pengguna.

3.4 Perancangan *Electrical* dan *Hardware* Alat Pemantau Kualitas Udara

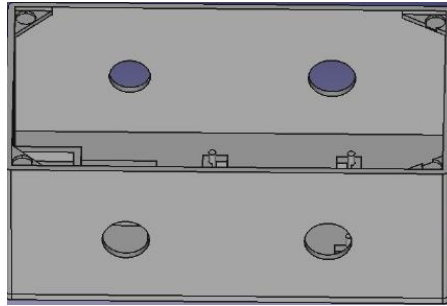
Pada tahap ini, sensor dan komponen yang digunakan dipertimbangkan saat merancang *electrical* dan *hardware* untuk Alat Pemantau Kualitas Udara. Mikrokontroler ESP-32 yang dilengkapi dengan berbagai sensor untuk mendeteksi kualitas udara seperti sensor gas, suhu, dan kelembapan adalah perangkat keras utama yang digunakan. Setelah penyusunan dan pengujian rangkaian elektrik, *hardware* diintegrasikan ke dalam sistem pemantauan untuk memaksimalkan pembacaan dan pengiriman data kualitas udara secara real-time melalui platform *Node-Red*.

3.4.1 Perancangan *Box* Alat

Dalam perancangan mekanik, dibuat struktur fisik yang akan menjadi tempat untuk semua komponen elektrik yang akan digunakan dalam alat ini. Perancangan mekanik dalam proyek ini berupa pembuatan *box* alat. *Box* alat dirancang cukup besar dengan mempertimbangkan ukuran dan bentuk dari PCB dan semua komponen yang digunakan namun tetap efisien sehingga mudah dibawa dan diletakkan. *Box* dibuat menggunakan material plastik menggunakan *3D printing* sehingga menjadikan *box* ini tahan terhadap debu dan kelembaban. *Box* berukuran ($p \times l \times t$) 100 mm×165 mm×100 mm dengan desain seperti pada Gambar 3.2 dan Gambar 3.3 di bawah ini:

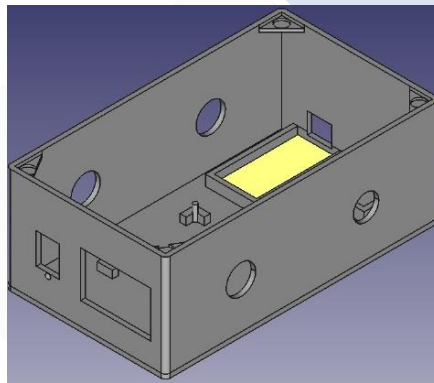


Gambar 3.2 Tampak Atas Desain *Box*



Gambar 3.3 Tampak Samping Desain *Box*

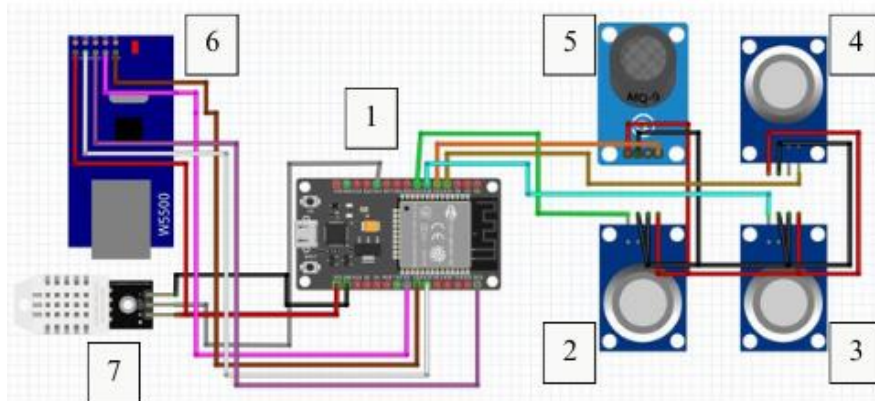
Desain *box* alat ini telah disesuaikan dengan semua komponen yang akan digunakan, sehingga lebih rapi dan dapat memaksimalkan kinerja setiap komponen. Maka, semua tempat peletakan komponen telah diatur, seperti Gambar 3.4 di bawah ini:



Gambar 3.4 Tampak Keseluruhan *Box*

3.4.2 Perancangan Elektrik

Perancangan elektrik menyangkut penyusunan dan penghubungan komponen elektronik yang akan digunakan untuk membuat kesatuan sistem pemantau yang dapat beroperasi secara efektif dan akurat. Perancangan elektrik dimulai dengan membuat skematik rangkaian menggunakan *software Fritzing*. Semua komponen seperti mikrokontroler, sensor dan *ethernet* modul dihubungkan, seperti yang terlihat pada Gambar 3.5 di bawah ini:



Gambar 3.5 Skematik Rangkaian Alat Pemantau Kualitas Udara

Keterangan:

- | | |
|-------------------------|-----------------|
| 1. Mikrokontroler ESP32 | 5. Sensor MQ-9 |
| 2. Sensor MQ-135 | 6. Modul W5500 |
| 3. Sensor MQ-8 | 7. Sensor DHT22 |
| 4. Sensor MQ-2 | |

3.5 Pembuatan dan Penggabungan Program dan *Hardware* Alat Pemantau Kualitas Udara

Pada tahap ini, semua komponen dan program yang telah diuji sebelumnya digabungkan menjadi satu sistem yang beroperasi. Langkah pertama pada tahap ini adalah memastikan bahwa semua komponen tersedia dan dalam kondisi baik. PCB yang telah dibuat berdasarkan desain sebelumnya telah siap untuk digunakan pada tahap ini. Selanjutnya, program yang telah dibuat menggunakan *Arduino IDE* dimasukkan ke dalam mikrokontroler ESP-32. Proses ini dimulai dengan memastikan bahwa semua kode dan *library* yang diperlukan telah diimpor dan diintegrasikan dengan benar ke dalam program. Kemudian, dengan menggunakan *software Arduino IDE*, program dapat diunggah ke ESP-32 melalui antarmuka USB. Proses ini memungkinkan ESP-32 menjalankan instruksi yang ditentukan oleh program, seperti inisialisasi sensor dan pengaturan komunikasi *Modbus* untuk mengirim data ke *server Node-Red*. Dengan demikian, ESP-32 berfungsi sebagai otak sistem pemantau kualitas udara, mengumpulkan data dari sensor,

memprosesnya dan mengirimkannya untuk dianalisis dan ditampilkan secara *real-time*.

3.6 Pengujian Alat Pemantau Kualitas Udara

Tahap ini bertujuan untuk memastikan bahwa alat pemantau kualitas udara berbasis *IoT* yang telah selesai dikerjakan dapat dioperasikan dan difungsikan berdasarkan spesifikasinya. Dimulai dengan menguji setiap sensor, semua sensor yang telah terhubung dengan mikrokontroler ESP32 diuji secara bersamaan untuk memastikan sensor – sensor ini dapat bekerja dengan sinkron tanpa hambatan. Setelah itu, dilakukan pengujian terhadap layar monitoring, ESP32 melakukan proses pengumpulan data dari semua sensor lalu mengirimkannya ke STB melalui protokol *Modbus*. Pengujian ini untuk memastikan tidak ada masalah pada data maupun pada komunikasi antar mikrokontroler dan STB yang dapat mengganggu operasi alat.

Selain itu, sistem Internet of Things (*IoT*) diuji untuk memastikan konektivitas dan keandalan pengiriman data secara *real-time*. Pengujian ini dilakukan dengan menggunakan STB untuk menghubungkan alat ke jaringan internet dan melihat bagaimana data dikirim dan ditampilkan pada *website* dan layar monitoring yang telah disiapkan.

3.7 Pembuatan Laporan Proyek Akhir

Proses terakhir dalam proyek akhir ini yaitu, pembuatan laporan proyek akhir. Pembuatan laporan proyek akhir berfungsi agar pendataan dari seluruh proses perancangan, pembuatan, perakitan hingga pengujian alat. Ada pula tujuan dari pembuatan laporan proyek akhir ini untuk menyusun bukti tertulis mengenai semua aspek dari latar belakang hingga kesimpulan.

BAB IV

PEMBAHASAN

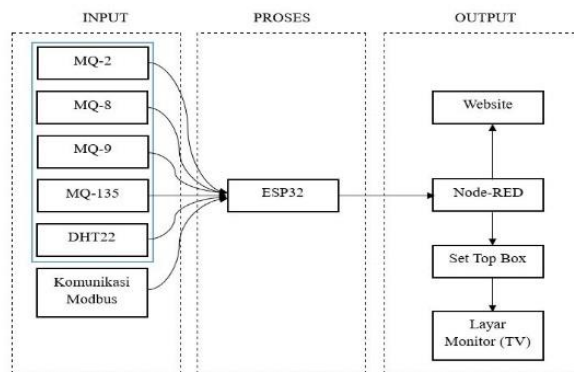
Berdasarkan metode yang diuraikan pada bab sebelumnya, maka pada bab ini akan dijelaskan mengenai alat, proses pengerjaan, pengujian hingga hasil akhir pada proyek akhir bertajuk “Alat Pemantau Kualitas Udara Berbasis *IoT*”.

4.1 Deskripsi Alat Pemantau Kualitas Udara Berbasis *IoT*

Alat pemantau kualitas udara berbasis *IoT* ini dirancang untuk mendeteksi berbagai macam parameter di udara, seperti konsentrasi gas berbahaya (metana, butana, hidrogen, karbon monoksida dan lainnya), dan suhu serta kelembaban. Maksud dasar dari alat ini adalah untuk memberikan sejumlah informasi tentang kondisi udara sekitar pengguna secara *real-time*. Alat ini menerapkan teknologi *Internet of Things (IoT)* guna memudahkan pemantauan melalui jaringan internet. Selain itu, alat ini juga dapat terhubung ke layar monitor dengan menggunakan STB melalui komunikasi *Modbus*. Dengan menggabungkan dua hal tersebut, Alat ini berperan dalam memantau dan melaporkan status kualitas udara dengan memberikan data yang akurat.

Alat ini menggunakan sensor MQ-2, MQ-8, MQ-9 dan MQ-135 untuk mendeteksi berbagai macam gas berbahaya. Sensor DHT22 digunakan untuk mengukur kelembaban dan suhu. Semua sensor di atas dihubungkan ke mikrokontroler ESP32 untuk mendeteksi masukan dari sensor dan mengirimkannya ke *website node-red* dan layar monitor.

4.2 Diagram Blok Alat Pemantau Kualitas Udara Berbasis *IoT*



Gambar 4.1 Diagram Blok Alat Pemantau Kualitas Udara Berbasis *IoT*

Berikut keterangan dari diagram blok di atas:

Blok *Input*

- Sensor MQ-2, sebagai pendeteksi gas LPG, metana, butana dan asap. Digunakan untuk mengidentifikasi keberadaan gas yang mudah terbakar dan asap di udara.
- Sensor MQ-8, sebagai pendeteksi gas hidrogen, sehingga dapat mencegah risiko terjadinya kebakaran dan ledakan.
- Sensor MQ-9, sebagai pendeteksi gas karbon monoksida (CO) dan gas yang mudah terbakar.
- Sensor MQ-135, sebagai pendeteksi dan pengukur berbagai gas polutan yang dapat mempengaruhi kualitas udara.
- Sensor DHT22, sebagai pengukur kelembaban dan suhu
- Komunikasi *Modbus*, sebagai protokol komunikasi serial yang memungkinkan transfer data antara mikrokontroler dan perangkat lainnya dalam sistem.

Blok Proses

- Mikrokontroler ESP32, sebagai pusat yang mengumpulkan, memproses dan mengirimkan data ke sistem *output*.

Blok *Output*

- Node-RED, sebagai *platform* untuk menampilkan hasil data secara *real-time* melalui *website* dan layar monitor (TV) yang melalui STB.

- *Website*, sebagai *interface* pengguna agar memudahkan akses melihat data kualitas udara seperti pada *smartphone*.
- *Set Top Box*, sebagai penghubung alat ke layar monitor (TV) agar memudahkan pemantauan di dalam ruangan.

4.3 Prinsip Kerja Alat Pemantau Kualitas Udara Berbasis *IoT*

Prinsip kerja alat pemantau kualitas udara berbasis *IoT* ini dimulai dari sensor mendeteksi berbagai macam parameter udara, seperti sensor MQ-2, MQ-8, MQ-9, dan MQ-135 mendeteksi berbagai macam gas dan DHT22 mendeteksi suhu dan kelembaban. Selanjutnya, Mikrokontroler ESP32 yang berfungsi sebagai pusat sistem, mengumpulkan data dari semua sensor, mengolahnya dan mengirimkannya ke *website* dan layar monitor melalui jaringan *WiFi* yang tersedia di ESP32.

Prinsip kerja *Internet of Things (IoT)* dalam proyek ini terdiri dari beberapa tahap penting. Pertama, sensor mengukur parameter lingkungan dan mengirimkan data ke ESP32. Kemudian, modul W5500 berfungsi sebagai *gateway* yang menghubungkan perangkat *IoT* ke *platform* pengolahan data *Node-RED*. Di *Node-RED*, data diproses dan ditampilkan dalam *dashboard* interaktif, yang dapat diakses oleh pengguna melalui *website* maupun layar TV. Kualitas udara dapat dipantau secara terus menerus dengan sistem ini.

4.4 Pembuatan Program Alat Pemantau Kualitas Udara

4.4.1 Pembuatan Program Mikrokontroler Alat Pemantau Kualitas Udara

Langkah selanjutnya adalah membuat program untuk mikrokontroler ESP32. Program ini bertanggung jawab untuk mengoperasikan semua sensor, mengumpulkan data dan mengirimkannya ke STB untuk pemrosesan selanjutnya, yaitu untuk menampilkan hasil sensor.

Untuk Pemrograman mikrokontroler, menggunakan *software Arduino*. Beberapa bagian penting dari program ini termasuk inisialisasi ESP32 dan semua sensor yang terhubung, pengumpulan data dari setiap sensor, pengolahan data mentah untuk memastikan keakuratannya, dan implementasi protokol *Modbus*

untuk mengkomunikasikan data antara ESP32 dan *Set Top Box* (STB), serta pengiriman data yang telah diolah ke STB melalui jaringan lokal.

4.5 Pembuatan dan Penggabungan Program dan *Hardware* Alat Pemantau Kualitas Udara

4.5.1 Pembuatan *Hardware* Mekanik

Pembuatan *hardware* mekanik merupakan tahap perakitan fisik dari semua komponen yang telah disiapkan. Semua komponen ditempatkan ke dalam *box* alat yang telah dirancang dan dicetak menggunakan 3D printing sebelumnya. Setiap komponen memiliki tempat dan ruang yang sesuai. Selain itu, dipasangkan magnet di setiap sudut bagian *box* dan tutup *box* untuk memastikan *box* dapat ditutup dengan kuat tetapi juga mudah dibuka saat diperlukan.

4.5.2 Pembuatan *Hardware* Elektrik

Pembuatan *hardware* elektrik merupakan tahap untuk memastikan semua komponen terhubung dengan benar dan siap beroperasi. Dalam proses ini, mikrokontroler ESP32, sensor MQ-2, MQ-8, MQ-9, MQ-135 dan DHT22, disambungkan ke PCB. Setiap komponen tersebut dihubungkan menggunakan konektor yang berupa kabel *jumper*.

4.5.3 Penggabungan Program dan *Hardware*

Penggabungan dimulai dengan menempatkan setiap komponen pada *box* alat sesuai dengan *layout* yang telah dirancang. Komponen yang telah terpasang disambungkan ke PCB menggunakan kabel. Semua sambungan tersebut wajib dikerjakan secara cermat agar tidak terjadi *short* ataupun sambungan yang longgar yang dapat mengganggu kinerja alat. Selain itu, kabel yang berlebihan diatur dan diikat dengan rapi untuk memastikan bahwa tidak ada kabel yang terlepas atau mengganggu fungsi alat.

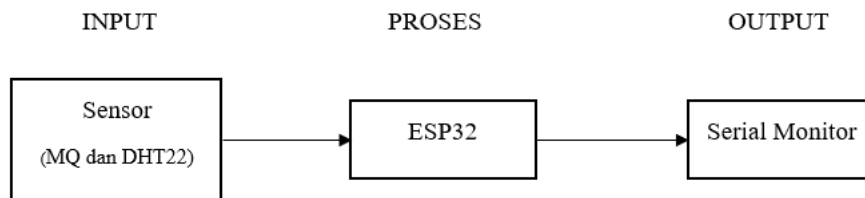
Setelah *hardware* dirakit, program *Arduino* diunggah ke ESP-32. Kemudian, fungsi *Modbus* ditambahkan ke dalam program *Arduino* untuk memungkinkan komunikasi antara ESP-32 dan *Node-Red*.



Gambar 4.2 Hasil Alat Pemantau Kualitas Udara

4.6 Pengujian Sensor

Pengujian sensor berguna untuk memastikan bahwa sensor yang akan digunakan dalam alat ini dapat beroperasi secara akurat dan optimal. Setiap sensor diuji secara terpisah untuk memastikan kinerjanya dalam mendeteksi parameter tertentu sesuai dengan spesifikasinya. Diagram blok pengujian sensor ditunjukkan, seperti:



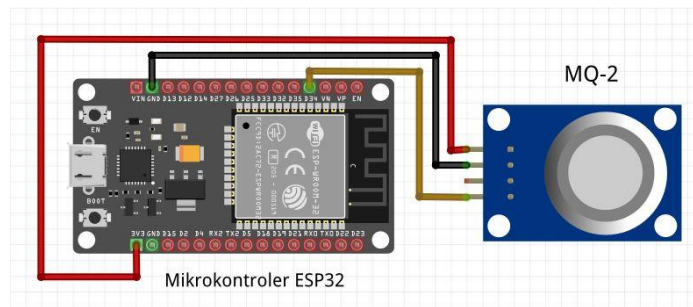
Gambar 4.3 Diagram Blok Pengujian Sensor

4.6.1 Pengujian Sensor MQ-2

Sensor MQ-2 mendeteksi gas yang mudah terbakar seperti LPG, metana, butana dan asap. Pengujian sensor mq-2 dilakukan menggunakan gas pemantik, hasil dari sensor dibandingkan dengan hasil dari *Air Quality Detector*. Pengujian dilakukan dengan beberapa tahapan.

4.6.1.1 Perancangan Elektrik Sensor MQ-2

Pada pengujian sensor MQ-2 ini sensor dihubungkan dengan ESP32 seperti skematik pada Gambar 4.4 berikut:



Gambar 4.4 Skematik Rangkaian Pengujian Sensor MQ-2

Setelah sensor dirangkai seperti skematik rangkaian di atas maka sensor akan diprogramkan dengan *code* utama:

```
int valueMQ2 = analogRead(MQ2_PIN);
```

Fungsi dari program ini adalah untuk membaca nilai tegangan analog dari sensor MQ-2. Setelah mendapatkan nilai tegangan analog, nilai tersebut akan diolah dengan menggunakan program :

```
float LPG_Concentration = (float)valueMQ2 / LPG_Ro *
LPG_Concentration_Ratio;
```

```
float CH4_Concentration = (float)valueMQ2 / CH4_Ro *
CH4_Concentration_Ratio;
```

Program *float LPG_Concentration* dan *float CH4_Concentration* adalah untuk memecah dan mengolah *raw value* dari nilai tegangan yang didapatkan sensor MQ-2 menjadi konsentrasi gas LPG dan CH₄.

4.6.1.2 Hasil Pengujian Sensor MQ-2

Setelah sensor dirangkai dan diprogramkan dengan *code* yang sesuai, maka sensor akan melakukan pengujian menggunakan gas pemantik. Hasil dapat terlihat dalam Tabel 4.1.

Tabel 4.1 Hasil Pengujian Sensor MQ-2

No.	Kondisi	Hasil Sensor MQ-2 (ppm)
1	Normal	Konsentrasi LPG: 160.75
		Konsentrasi Metana: 214.33
2	Gas pemantik dinyalakan tanpa api	Konsentrasi LPG: 267.25
		Konsentrasi Metana: 356.33
3	Gas pemantik dinyalakan dengan api	Konsentrasi LPG: 180.75
		Konsentrasi Metana: 241.00

Gas pemantik berisikan butana, propana ataupun LPG (terdiri dari butana dan propana). Metana tidak berasal langsung dari gas pemantik, tetapi karena sensitivitas sensor MQ-2 terhadap metana menyebabkan sensor ini dapat mendeteksi metana dalam gas – gas hidrokarbon yang dilepaskan dari gas pemantik pada pengujian.

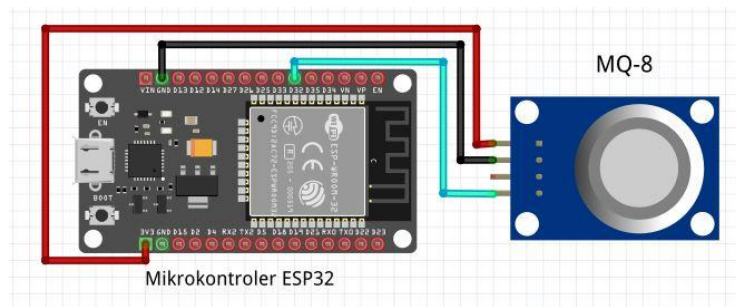
Ketika gas pemantik dinyalakan tanpa api maka yang dihasilkan hanya gas murni tanpa terjadinya pembakaran, menyebabkan sensor mendeteksi lebih banyak konsentrasi gas LPG dan metana seperti yang terdapat pada poin 2. Pada poin 3, gas pemantik mengeluarkan lebih sedikit konsentrasi gas LPG dan metana di udara karena terjadinya pembakaran sehingga menghasilkan lebih sedikit gas yang dideteksi.

4.6.2 Pengujian Sensor MQ-8

Sensor MQ-8 mendeteksi gas hidrogen, sensor ini digunakan untuk mendeteksi konsentrasi hidrogen dalam atmosfer. Pengujian sensor MQ-8 dapat dilakukan dengan beberapa tahapan.

4.6.2.1 Perancangan Pengujian Sensor MQ-8

Pengujian sensor MQ-8 dimulai dengan membuat perancangan pada sensor, rancangannya yaitu rancangan skematik rangkaian yang digunakan sebagai dasar untuk merangkai sensor dan ESP32. Berikut skematik rangkaian sensor MQ-8 pada Gambar 4.5.



Gambar 4.5 Skematik Rangkaian Pengujian Sensor MQ-8

Lalu sensor dihubungkan dengan ESP32 seperti skematik rangkaian di atas. Setelah dihubungkan, mikrokontroler diprogramkan dengan *code* utama:

```
int valueH2 = analogRead(MQ8_PIN);
```

Fungsi dari program ini adalah untuk membaca nilai tegangan analog dari sensor MQ-8. Setelah mendapatkan nilai tegangan analog, nilai tersebut akan diolah dengan menggunakan program :

```
float ratioH2 = (float)valueH2 / H2_Ro;
```

```
float H2_Concentration = H2_Kc * pow(ratioH2, -1.3);
```

Program *float ratioH2* adalah untuk mendapatkan nilai rasio dari konsentrasi H₂ dan *float H2_Concentration* adalah untuk memecah dan mengolah *raw value* dari nilai tegangan yang didapatkan sensor MQ-2 menjadi konsentrasi gas H₂.

4.6.2.2 Hasil Pengujian Sensor MQ-8

Setelah mengikuti tahap sebelumnya, maka sensor akan diuji dengan gas pemantik. Kemudian hasil dari pengujian akan dimasukkan ke dalam Tabel 4.2.

Tabel 4.2 Hasil Pengujian Sensor MQ-8

No.	Kondisi	Hasil Sensor MQ-8 (ppm)
1	Normal	0.03
2	Gas pemantik dinyalakan tanpa api	0.02
3	Gas pemantik dinyalakan dengan api	0.03

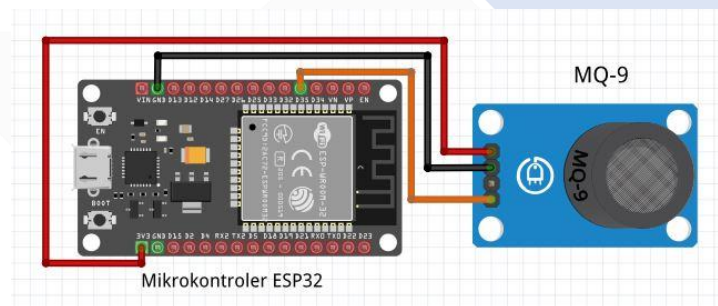
Gas yang dikeluarkan dari gas pemantik dapat mengubah pola aliran udara di sekitar sensor MQ-8. Seperti data hasil pengujian pada Tabel 4.2, saat gas pemantik dinyalakan maka kadar H_2 yang terdeteksi oleh sensor MQ-8 pun lebih sedikit ketika dalam keadaan normal.

4.6.3 Pengujian Sensor MQ-9

Sensor MQ-9 mendeteksi gas karbon monoksida (CO) dan gas yang mudah terbakar lainnya. Pada pengujian sensor MQ-9, sensor diuji menggunakan gas pemantik dan hasilnya akan dibandingkan dengan *Air Quality Detector*. Adapun berikut tahapan melakukan pengujian sensor MQ-9.

4.6.3.1 Perancangan Elektrik Pengujian Sensor MQ-9

Pengujian sensor MQ-9 diawali dengan menghubungkan sensor dengan mikrokontroler seperti skematik pada Gambar 4.6 berikut:



Gambar 4.6 Skematik Rangkaian Pengujian Sensor MQ-9

Setelah sensor dirangkai seperti skematik rangkaian di atas maka sensor akan diprogramkan dengan *code* utama:

```
int valueMQ9 = analogRead(MQ9_PIN);
```

Fungsi dari program ini adalah untuk membaca nilai tegangan analog dari sensor MQ-9. Setelah mendapatkan nilai tegangan analog, nilai tersebut akan diolah dengan menggunakan program :

```
float MQ9_sensor_volt = ((float)valueMQ9 / 4095) * MQ9_V_RL;
```

`float MQ9_sensor_volt` digunakan untuk mengubah nilai tegangan analog yang didapatkan oleh sensor menjadi sinyal digital.

```
float MQ9_RS_gas = ((MQ9_V_RL * MQ9_RL) / MQ9_sensor_volt) - MQ9_RL;
```

Jika tegangan analog telah diperoleh, langkah selanjutnya adalah menghitung resistansi yang didapatkan oleh sensor.

```
float ratioMQ9 = MQ9_RS_gas / MQ9_Ro; // Rasio Rs/Ro
float CO_concentration = pow(10, ((log(ratioMQ9) - CO_curve[1]) /
CO_curve[2]) + CO_curve[0]); // Menggunakan interpolasi log-log untuk
mencari PPM
```

Setelah mendapatkan nilai resistansi sensor, fungsi dari *float ratioMQ9* adalah untuk menghitung konsentrasi gas CO dengan kurva karakteristik menggunakan interpolasi log-log.

4.6.3.2 Hasil Pengujian Sensor MQ-9

Setelah sensor dirangkai dan diprogramkan maka dilakukan pengujian dengan menggunakan gas pemantik, hasil dari pengujian terdapat di dalam Tabel 4.3.

Tabel 4.3 Hasil Pengujian Sensor MQ-9

No.	Kondisi	Hasil Sensor MQ-9 (ppm)	Hasil <i>Air Quality Detector</i> (ppm)	Persentase Error (%)
1	Normal	1.02	1	2
2	Gas pemantik dinyalakan tanpa api	5.2	5	4
3	Gas pemantik dinyalakan dengan api	4.1	4	2.5
Rata – Rata Error (%)				2.83

Persentase eror di atas dihitung dengan menggunakan rumus:

$$\text{Persentase Error (\%)} = \left| \frac{\text{Hasil Sensor} - \text{Hasil Alat Pembanding}}{\text{Hasil Alat Pembanding}} \right| 100 \dots\dots\dots 4.3$$

$$\text{Persentase Error (\%)} = \left| \frac{1.02 - 1}{1} \right| 100 = 2\%$$

Setelah didapatkan persentase erornya maka selanjutnya dapat menghitung rata – rata eror sensor menggunakan rumus berikut:

$$\text{Rata - Rata Error (\%)} = \frac{\sum \text{Persentase Error}}{\sum \text{Data}} \dots\dots\dots 4.3$$

$$\text{Rata - Rata Error (\%)} = \frac{\sum 2 + 4 + 2.5}{3} = 2.83$$

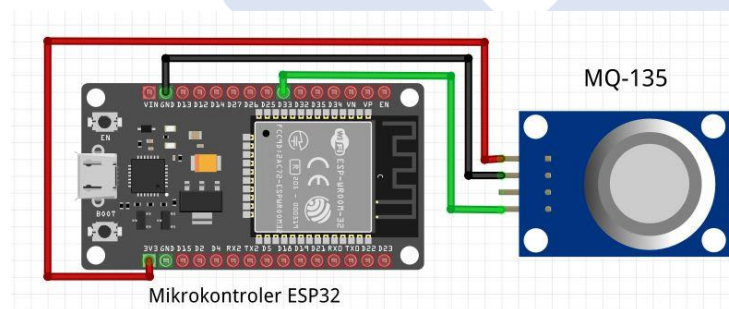
Dari hasil pengujian pada sensor MQ-9 Sensor MQ-9 sangat peka terhadap gas pemantik, baik ketika dinyalakan tanpa api maupun dengan api. Nilai yang dicatat oleh sensor dalam kedua kondisi ini menunjukkan bahwa sensor dapat mendeteksi perubahan konsentrasi gas berbahaya di udara. Persentase eror sensor MQ-9 yang mendapati hasil 2.83%. Ini menunjukkan bahwa sensor MQ-9 sangat efektif dalam melacak kualitas udara, terutama untuk mendeteksi gas beracun seperti karbon monoksida dan gas pemantik.

4.6.4 Pengujian Sensor MQ-135

Sensor MQ-135 mendeteksi berbagai jenis gas beracun seperti karbon dioksida (CO₂). Selain itu, dengan sensor MQ-135 dapat diketahui juga kategori kualitas udaranya dengan satuan pengukuran *parts per million* (ppm). Ada pula berikut merupakan tahapan pengujian sensor MQ-135.

4.6.4.1 Perancangan Pengujian Sensor MQ-135

Untuk melakukan pengujian sensor MQ-135, sensor harus dihubungkan ke mikrokontroler dengan mengikuti skematik pada Gambar 4.7 berikut:



Gambar 4.7 Skematik Rangkaian Pengujian Sensor MQ-135

Setelah sensor dirangkai seperti skematik rangkaian di atas maka sensor akan diprogramkan dengan *code* utama:

```
int MQ135Value = analogRead(MQ135_PIN);
```

Fungsi dari program ini adalah untuk membaca nilai tegangan analog dari sensor MQ-135. Setelah mendapatkan nilai tegangan analog, nilai tersebut akan diolah dengan menggunakan program :

$$\text{float MQ135_voltage} = \text{MQ135Value} * (3.3 / 4095.0);$$

Jika tegangan analog telah diperoleh, langkah selanjutnya adalah menghitung resistansi yang didapatkan oleh sensor.

$$\text{float MQ135_RS_air} = ((3.3 * 10.0) / \text{MQ135_voltage}) - 10.0;$$

Setelah mendapatkan nilai resistansi sensor, *float MQ135_ratio* berfungsi untuk mendapatkan nilai rasio dari sensor MQ-135.

$$\text{float MQ135_ratio} = \text{MQ135_RS_air} / \text{MQ135_RZERO};$$

Setelah itu, nilai rasio tersebut akan digunakan untuk mendapatkan konsentrasi CO₂ dengan menggunakan parameter dari gas CO₂.

$$\text{ppmco2} = \text{CO2_PARA} * \text{pow}(\text{MQ135_ratio}, -\text{CO2_PARB});$$

4.6.4.2 Hasil Pengujian Sensor MQ-135

Selanjutnya sensor akan diuji dengan ditiup sebanyak 3 kali untuk mengukur tingkat CO₂ di sekitar sensor. Hasil dari sensor dapat dilihat dalam tabel 4.4.

Tabel 4.4 Hasil Pengujian Sensor MQ-135

No.	Kondisi	Hasil Sensor MQ-135 (ppm)	Hasil <i>Air Quality Detector</i> (ppm)	Persentase Error (%)
1	Normal	283	400	<i>invalid</i>
2	Gas pemantik dinyalakan tanpa api	1844	1767	4.4
3	Gas pemantik dinyalakan dengan api	917	851	7.8
Rata – Rata Error (%)				6.1

Air Quality Detector yang digunakan memiliki *range* 400 – 5000 ppm, maka *Air Quality Detector* tidak dapat membaca konsentrasi gas yang di bawah

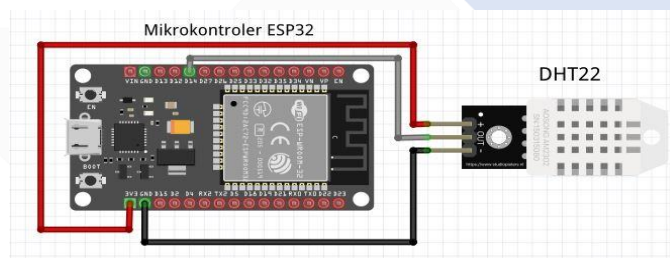
range-nya sehingga nilai eror pada kondisi normal tidak diketahui. Rata – rata eror didapatkan dari dua data saja yaitu, data saat gas pemantik dinyalakan tanpa api dan dengan api, rata – rata yang didapatkan sebesar 6.1%. Dari nilai rata – rata eror ini maka sensor MQ-135 menghasilkan data yang cukup baik.

4.6.5 Pengujian Sensor DHT22

Sensor DHT22 digunakan untuk mengukur suhu dan kelembaban. Keluaran suhu yang diukur menggunakan satuan derajat *celsius* (°c) dan untuk kelembaban menggunakan persen (%). Berikut tahapan pengujian sensor DHT22.

4.6.5.1 Perancangan Pengujian Sensor DHT22

Perancangan pengujian sensor DHT22 merupakan tahap awal untuk melakukan pengujian ini. Berikut merupakan skematik rangkaian sensor DHT22 yang menjadi dasar untuk merangkai sensor dan mikrokontroler pada Gambar 4.8.



Gambar 4.8 Skematik Rangkaian Pengujian Sensor DHT22

Dengan skematik di atas, maka sensor akan dihubungkan dengan mikrokontroler. Setelah itu, mikrokontroler akan diprogramkan dengan *code* utama:

```
dht.begin();
```

Fungsi program *dht.begin()* adalah untuk menginisialisasi sensor DHT-22 menggunakan library DHT-22.

```
float t = 0.0;
```

```
float h = 0.0;
```

```
t = dht.readTemperature();
```

```
h = dht.readHumidity();
```

dht.readTemperature() berfungsi untuk membaca nilai dari suhu dan *dht.readHumidity()* berfungsi untuk membaca nilai dari kelembaban.

4.6.5.2 Hasil Pengujian Sensor DHT22

Sensor yang telah dihubungkan dengan mikrokontroler yang telah diprogramkan tadi akan diuji dengan menggunakan panas dari lilin menyala. Adapun hasil dari sensor DHT22 seperti yang terdapat dalam Tabel 4.5 berikut.

Tabel 4.5 Hasil Pengujian Sensor DHT22

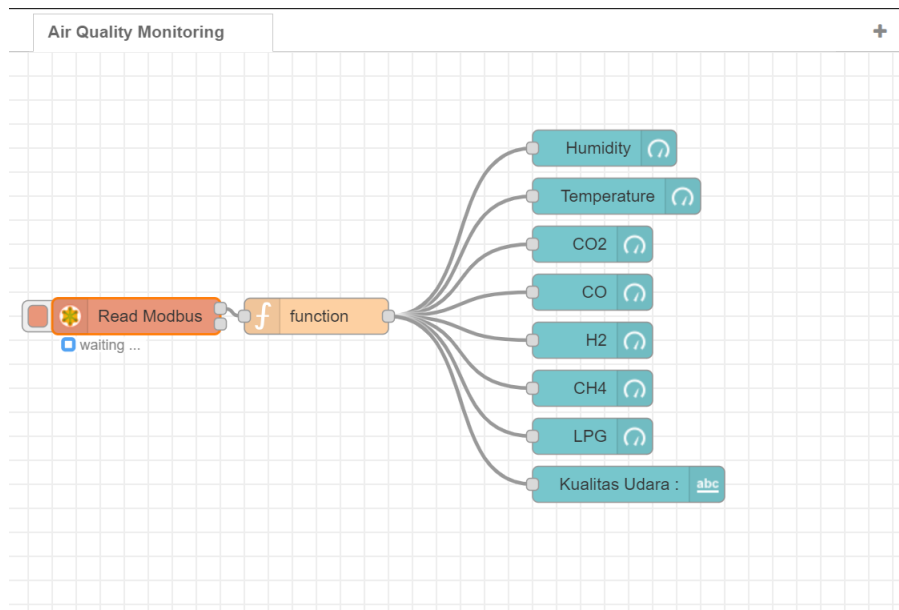
No	Kondisi	Hasil Sensor DHT22	Hasil <i>Air Quality</i> <i>Detector</i>	Persentase Error (%)
1	Normal	Suhu: 33.9°C	Suhu: 34.1°C	0.59
		RH: 72.1%	RH: <i>invalid</i>	<i>invalid</i>
2	Api lilin dinyalakan	Suhu: 43.3°C	Suhu: 43.3°C	0
		RH: 45.6%	RH: <i>invalid</i>	<i>invalid</i>
3	Setelah api lilin dimatikan	Suhu: 38.6°C	Suhu: 38.9°C	0.77
		RH: 57.9%	RH: <i>invalid</i>	<i>invalid</i>
Rata – Rata Error (%)				0.5

Berdasarkan hasil pengujian di atas, didapatkan rata-rata eror untuk pengukuran suhu adalah 0.5%. Hasil pengujian suhu sensor DHT22 dibandingkan dengan hasil dari termometer ruangan. Pada pengujian ini untuk kelembaban tidak dihitung karena pembandingnya hanya menggunakan pembandingan data kelembaban wilayah, sehingga perbedaannya pun tidak spesifik. Namun, udara yang lebih dingin tidak dapat menahan kelembapan sebanyak udara yang lebih hangat, karena pada saat suhu meningkat maka semakin banyak uap air yang dapat ditampung di udara [33]. Pada pengujian ini saat suhu meningkat atau lebih hangat maka *relative humidity* (RH) pun menurun dapat dilihat pada poin 2.

4.7 Pengujian *Node-Red*

Node-Red digunakan untuk menghubungkan ESP-32 ke server *hosting Node-Red* melalui komunikasi *Modbus*. Setelah terhubung ke server *Node-Red*, ESP-32 akan mengirimkan data sensor melalui koneksi *Modbus* ke situs *web Node-Red*.

4.7.1 Flow Node-Red



Gambar 4.9 Flow Node-Red

Untuk mengirim data dari ESP-32, tambahkan alamat *IP*-nya ke *menu (Host)* di *Node (Read Modbus)*. *Node (Read Modbus)* membaca nilai yang dikirim oleh ESP-32 menggunakan koneksi *Modbus*. Setelah ESP-32 terhubung ke server *Node-Red*, data harus diproses menggunakan *Node (function)*. Program harus ditulis ke *Node (Function)* agar data yang dikirim oleh ESP-32 dapat dibaca di server *Node-Red*. Data yang diproses akan ditampilkan menggunakan *Node (Gauge)* dan (*Text*).

4.8 Pengujian Keseluruhan Alat Pemantau Kualitas Udara Berbasis *IoT*

Tahap ini untuk memastikan bahwa keseluruhan sistem dapat beroperasi dengan baik dan akurat. Pengujian dilakukan di berbagai kondisi untuk melihat kesesuaian hasil dari sensor yang tampil di layar monitor dan *website*. Berikut tahap – tahap yang akan dilakukan untuk melakukan pengujian ini.

4.8.1 Hasil Pengujian Keseluruhan Sistem dan Alat

Setelah semua komponen dirangkai dan diprogramkan, maka kemudian dilakukan pengujian dengan lima kondisi yaitu:

1. Normal;
2. Lilin menyala;
3. Asap hasil pembakaran kertas;
4. Asap hasil pembakaran sampah plastik.

Dengan keempat kondisi tersebut maka hasil data dari sensor dapat dilihat melalui *website* maupun layar monitor yang tersambung dengan STB seperti layar TV pada Gambar 4.10 dan Gambar 4.11 sebagai berikut:



Gambar 4.10 Tampilan *Website*



Gambar 4.11 Tampilan Layar TV

Tabel 4.6 Hasil Pengujian Keseluruhan Sistem dan Alat

Kondisi	Nilai Parameter							Kategori Kualitas Udara
	LPG (ppm)	CH ₄ (ppm)	H ₂ (ppm)	CO (ppm)	CO ₂ (ppm)	Suhu (°C)	RH (%)	
Normal	169.2	226.3	0.03	0.18	135.6	28	93	Baik
Lilin menyala	210.7	281	0.01	0.18	209.9	31.3	88.2	Baik
Asap sampah kertas	263.3	351	0.01	3.25	1478	31	75.6	Sangat Tidak Sehat
Asap sampah plastik	257.3	343	0.01	1.04	1164.4	30.7	76.2	Tidak Sehat

Hasil pengujian alat di atas akan dibandingkan dengan hasil pengujian pada *air quality detector*. Namun, *air quality detector* tidak memiliki indikator pengujian untuk LPG, CH₄ dan H₂, maka dari itu parameter yang dibandingkan hanya CO, CO₂ dan suhu. Untuk kelembaban menggunakan nilai kelembaban wilayah, jadi kelembapannya pun tidak memiliki perbedaan yang signifikan. Data hasil dari *air quality detector* terdapat pada Tabel 4.7.

Tabel 4.7 Nilai Hasil *Air Quality Detector*

Kondisi	Nilai Parameter							Kategori Kualitas Udara
	LPG (ppm)	CH ₄ (ppm)	H ₂ (ppm)	CO (ppm)	CO ₂ (ppm)	Suhu (°C)	RH (%)	
Normal	-	-	-	1	400	27	-	<i>Good</i>
Lilin menyala	-	-	-	1	400	30.7	-	<i>Slight</i>
Asap sampah kertas	-	-	-	3	1422	30.9	-	<i>Serious</i>
Asap sampah plastik	-	-	-	1	1085	30.1	-	<i>Serious</i>

Adapun berikut persentase rata – rata eror untuk pengujian di atas. CO sebesar 6.15%; CO₂ sebesar 5.61%; Suhu sebesar 2%. Dikarenakan rentang pada *air quality detector*, yaitu rentang CO 1 – 999 ppm dan CO₂ 400 – 5000 ppm maka nilai CO dan CO₂ yang didapatkan hanya perbandingan dari data pada poin 3 dan 4. Selain itu, data yang dihasilkan pada *air quality detector* tidak menampilkan

angka desimal sehingga nilai rata – rata eror yang dihasilkan lumayan tinggi meskipun perbedaan hasil dari kedua alat sangat kecil. Nilai rata – rata eror pada parameter tersebut masih dalam batas akurasi yang dapat diterima.

Konsentrasi LPG, CH₄ dan H₂ tidak memiliki indikator pembanding, tetapi dari hasil pengujian tiap sensor, sensor yang digunakan untuk mendeteksi parameter – parameter tersebut tetap menghasilkan data yang cukup baik. Untuk kelembaban pembanding yang tersedia hanya menampilkan data kelembaban wilayah setempat sehingga tidak memiliki perbedaan yang signifikan.

Hasil pengujian ini menunjukkan bahwa alat pemantau kualitas udara berbasis *IoT* ini dapat mendeteksi dan mengukur faktor-faktor penting untuk kualitas udara. Persentase rata – rata eror pada suhu menunjukkan bahwa sensor menghasilkan data dengan akurasi yang baik. Untuk parameter gas CO dan CO₂ persentase rata-rata kesalahan yang diperoleh menunjukkan bahwa sensor yang digunakan memiliki tingkat akurasi yang cukup baik. Hasil pengujian menunjukkan bahwa sensor-sensor yang mendeteksi LPG, CH₄, H₂, dan kelembaban bekerja dengan baik. Secara keseluruhan, tujuan proyek ini untuk menyediakan solusi pemantauan kualitas udara yang efisien dan efektif telah dicapai.

BAB V

PENUTUP

5.1 Kesimpulan

Didasarkan atas hasil pengujian yang telah diperoleh dari proyek akhir Alat Pemantau Kualitas Udara Berbasis *IoT* ini menunjukkan bahwa:

1. Perangkat yang terintegrasi dengan *IoT* ini berfungsi dengan rata-rata eror seluruh sensor 4.6% dan memenuhi tujuannya untuk menyediakan data tentang kualitas udara secara *real-time* dan daring.
2. Dengan pengujian menggunakan berbagai sumber polusi, seperti api lilin, asap sampah kertas dan plastik, alat ini dapat mendeteksi rata – rata eror konsentrasi gas seperti CO sebesar 6.15%; CO₂ sebesar 5.61%; suhu sebesar 2%. Selain itu, pengujian pada gas LPG, CH₄, H₂, serta kelembaban juga dapat mendeteksi perubahan parameter pada berbagai sumber polusi.
3. Pada pengujian ini menunjukkan bahwa alat ini dapat menampilkan data yang sesuai pada *website Node-RED* dengan kecepatan pengiriman data *real-time* sebesar satu detik.

5.2 Saran

Setelah merampungkan proyek akhir ini, didapatkan sejumlah saran mengenai pengembangan selanjutnya dari alat pemantau kualitas udara ini, yaitu sebagai berikut:

1. Membuat data dapat tersimpan di dalam *database* sehingga memungkinkan untuk membandingkan data yang dihasilkan alat.
2. Menggunakan sensor yang memiliki tingkat akurasi dan sensitivitas yang lebih tinggi.
3. Menggunakan teknologi *IoT* yang dapat memungkinkan pengiriman data dari lokasi yang lebih jauh dan tidak menggunakan internet yang sama.

DAFTAR PUSTAKA

- [1] Ulaan, G. C., Poekoel, V. C., & Ontowirjo, A. H. (2022). *Indoor Air Quality Monitoring System*. Jurnal Teknik Informatika.
- [2] Kinerja Pengendalian Pencemaran dan Kerusakan Lingkungan Tahun 2023. (2023, Desember 29). Diakses pada 18 Juni 2024, dari Kementerian Lingkungan Hidup dan Kehutanan: <https://ppid.menlhk.go.id/berita/siaran-pers/7574/kinerja-pengendalian-pencemaran-dan-kerusakan-lingkungan-tahun-2023>
- [3] Rosalina, M. P., Wisanggeni, S. P., & Krisna, A. (2023, September 23). Alat Pemantauan Terbatas, Data yang Diperoleh Tidak Ideal. Diakses pada 16 Juni 2024, dari Kompas: <https://www.kompas.id/baca/investigasi/2023/09/21/buta-tanpa-data-kualitas-udara>.
- [4] Chaniago, D., Zahara, A., & Ramadhani, I. S. (2020, September 24). Indeks Standar Pencemar Udara (ISPU) Sebagai Informasi Mutu Udara Ambien di Indonesia. Diakses pada 2 Mei 2024, dari Direktorat Pengendalian Pencemaran Udara: <https://ditppu.menlhk.go.id/portal/read/indeks-standar-pencemar-udara-ispu-sebagai-informasi-mutu-udara-ambien-di-indonesia>.
- [5] Prayudha, J., Pranata, A., & Hafiz, A. A. (2018). Implementasi Metode *Fuzzy Logic* untuk Sistem Pengukuran Kualitas Udara di Kota Medan Berbasis *Internet Of Things (IOT)*. JURTEKSI (Jurnal Teknologi Dan Sistem Informasi) Vol. IV No. 2, Jun 2018, Hlm. 141 – 148.
- [6] Liandy, A. (2018). Rancang Bangun Pemantauan Gas Berbahaya Dan Suhu Pada Ruangan Melalui Website Berbasis Arduino. Institut Teknologi Nasional Malang.
- [7] *Permissible Exposure Limits – Annotated Tables*. Diakses pada 18 Juni 2024, dari *Occupational Safety and Health Administration*: <https://www.osha.gov/annotated-pels/table-z-1#>

- [8] *Methane: Health And Safety Hazards Fact Sheet*. (2021, January 5). Diakses pada 22 Mei 2024, dari mineARC: <https://minearc.com/methane-health-and-safety-hazards-fact-sheet/>
- [9] *OSHA Occupational Chemical Database*. (2021, January 29). Diakses pada 22 Mei 2023, dari *Occupational Safety and Health Administration*: <https://www.osha.gov/chemicaldata/563>
- [10] *Carbon Monoxide*. (2024, March 14). Diakses pada 23 Mei 2024, dari *United States Environmental Protection Agency*: <https://www3.epa.gov/region1/airquality/co.html>
- [11] *CO₂ Exposure Limits – Is Your Workplace And Draft Beer System Compliant?*. Diakses pada 23 Mei 2024, dari *Micro Matic*: <https://www.micromatic.com/en-us/learn/dispensing-knowledge/learning-resource-center/co2-permissible-exposure-limits-health-hazard-safety>
- [12] Kelembapan Udara Ideal:Faktor Kunci untuk Kesehatan dan Kesejahteraan. (2023, May 31). Diakses pada 18 Juni 2024, dari *Higienis Indonesia*: <https://www.higienis.com/blog/kelembapan-udara-ideal-faktor-kunci-untuk-kesehatan-dan-kesejahteraan/>
- [13] *The Ideal Room Temperature for Every Situation & 7 Ways to Maintain It!* (2024, March 8). Diakses pada 22 Mei 2024, dari *CIELO*: <https://cielowigle.com/blog/ideal-room-temperature/>
- [14] Indeks Standar Pencemar Udara (ISPU). (2023, September 22). Diakses pada 20 Juni 2024, dari Dinas Komunikasi dan Informatika Provinsi Jawa Barat: <https://arcgis.jabarprov.go.id/portal/apps/storymaps/stories/5cbe0dd241cc4630bb2ce92eabf3772c>
- [15] Day, B. (2019, August). Kualitas Udara Memburuk, Pahami Indeks Standar Pencemar Udara (ISPU) dan Dampaknya bagi Kesehatan. Diakses pada 20 Juni 2024, dari *Climate*: https://www.climate4life.info/2019/08/kualitas-udara-memburuk-pahami-indeks-standar-pencemar-udara-dan-dampaknya-pada-kesehatan.html#google_vignette

- [16] Anjani, N. (2024, January 9). Apa Itu *Internet of Things*? Pengertian, Cara Kerja, dan Manfaatnya. Diakses pada 15 Juni 2024 DCloud: <https://dcloud.co.id/blog/apa-itu-internet-of-things.html>
- [17] Rumampuk, G. C., Poekoel, V. C., & Rumagit, A. M. (2021). *Internet of Things-Based Indoor Air Quality Monitoring System Design*. Jurnal Teknik Informatika vol.17 no. 1 January – March 2021, pp. 11-18.
- [18] Rombang, I. A., Setyawan, L. B., & Dewantoro, G. (2022). Perancangan Prototipe Alat Deteksi Asap Rokok dengan Sistem Purifier Menggunakan Sensor MQ-135 dan MQ-2.
- [19] Frans. (2022, January 28). 18 Jenis Sensor Gas seri MQ. Diakses pada 15 Mei 2024, dari Anak Teknik: <https://www.anakteknik.co.id/rahasia1/articles/18-jenis-sensor-gas-seri-mq>
- [20] Sohibun, Daruwati, I., Hatika, R., & Mardiansyah, D. (2021). *MQ-2 Gas Sensor using Micro Controller Arduino Uno for LPG Leakage with Short Message Service as a Media Information*. *Journal of Physics: Conference*.
- [21] Gunawan, D., Margono, & Sudrajat. (2019). Detektor Gas Menggunakan Sensor Mq-9 Berbasis Mikrokontroler Arduino Uno Di Politeknik Penerbangan Surabaya. *Prosiding SNITP*.
- [22] Sunny, A. I., Zhao, A., Li, L., & Sakiliba, S. K. (2021). *Low-Cost IoT-Based Sensor System: A Case Study on Harsh Environmental Monitoring*. MDPI.
- [23] Erik, M., Nurdiyanto, F., & Hidayat, R. (2024). AeroSenseMonitor: Integrasi Sensor DHT11 dan MQ135 untuk Pemantauan Kualitas Udara Berbasis Arduino Uno. *Jurnal Komputer dan Elektro Sains*, Vol.2 No.2, September 2024, 8-11.
- [24] Nari, M. I., Laksono, J. A., Hidayat, B. U., & Udin, A. R. (2021). *The Portable Carbon Monoxide (CO) and Hydrocarbons (HC) Gas Detection System in The Car Cabin Uses The MQ-9 and TGS 2610 Sensors*. *Indonesian Journal Of Engineering Research*
- [25] Musbikhin. (2020, September 9). Apa itu Sensor DHT11 Dan DHT22 serta Perbedaannya. Diakses pada 24 Mei 2024, dari

<https://www.musbikhin.com/apa-itu-sensor-dht11-dan-dht22-serta-perbedaannya/>

- [26] Nugraha, R. F., Husna, F. N., Sand, Syahla, A. F., Saputra, Y. A., & Hidayat, R. (2024). *Smart Air Quality Guardian: Pengawasan Polusi Udara Berbasis ESP32 dengan Sensor Gas MQ-2 dan MQ-135*. Jurnal Komputer dan Elektro Sains.
- [27] Gheorge, A., & Stoica, C. (2021). *Wireless Weather Station Using Arduino Mega And Arduino Nano*. *Scientific Bulletin of the Electrical Engineering Faculty*.
- [28] Sukindar. (2022, March 9). Apa Itu Set Top Box yang Bisa Dipakai Untuk Menonton Siaran TV Digital?. Diakses pada 30 Mei 2024, dari Gadgetren: <https://gadgetren.com/2022/03/09/mengenal-apa-itu-set-top-box-160889/>
- [29] Sihombing, Y. A., & Listiari, S. (2020). *Detection of Air Temperature, Humidity and Soil pH by Using DHT22 and pH Sensor Based Arduino Nano Microcontroller*. *AIP Conference Proceedings*.
- [30] Iyalunaidu, S. (2024, January 25). *Modbus Communication Protocol*. Diakses pada 30 Mei 2024, dari AutomationForum.co: <https://automationforum.co/Modbus-communication-protocol/>
- [31] Innes, B. (2021, June 14). *Develop production-ready, containerized Node-RED apps*. Diakses pada 17 Juli 2024, dari IBM: <https://developer.ibm.com/learningpaths/develop-containerized-node-red-apps/>
- [32] Shakir, N. (2023, March 6). *Turn Your Smartphone Into an IoT Device*. Diakses pada 17 Juli 2024, dari IBM: <https://developer.ibm.com/tutorials/iot-mobile-phone-iot-device-bluemix-apps-trs/>
- [33] *What Is Humidity*. Diakses pada 18 Juni 2024, dari *Airthings*: <https://www.airthings.com/what-is-humidity>



LAMPIRAN

LAMPIRAN 1

Daftar Riwayat Hidup (Perorangan)

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Muhammad Abdurahman Alfarizi
Tempat & Tanggal Lahir : Pekanbaru, 27 Juni 2003
Alamat Rumah : Jalan Depati Amir Nomor 13A
No. HP : 0812 1120 1188
Email : alfarizi0856@gmail.com
Jenis Kelamin : Laki-laki
Agama : Islam



2. Riwayat Pendidikan

1. MIN 1 Pekanbaru	Lulus 2015
2. SMPN 1 Pemali	Lulus 2018
3. SMAN 1 Pemali	Lulus 2021
4. Politeknik Manufaktur Negeri Bangka Belitung	2021 – Sekarang

3. Pendidikan Non-Formal

Praktik Kerja Lapangan di PT *Chronos Universal*.

Sungail Liat, 25 Juli 2024

Muhammad Abdurahman Alfarizi

DAFTAR RIWAYAT HIDUP

4. Data Pribadi

Nama Lengkap : Tiara Yasmin Arimbi Chantika
Tempat & Tanggal Lahir : Palangka Raya, 2 Mei 2003
Alamat Rumah : Jalan Mendanau 2 Nomor 27
No. HP : 0897 1154 180
Email : tiarayasmin2503@gmail.com
Jenis Kelamin : Perempuan
Agama : Islam



5. Riwayat Pendidikan

- | | |
|---|-----------------|
| 1. SDN 3 Menteng | Lulus 2014 |
| 2. SMPN 3 Palangka Raya | Lulus 2017 |
| 3. MAN 2 Yogyakarta | Lulus 2020 |
| 4. Politeknik Manufaktur Negeri Bangka Belitung | 2021 – Sekarang |

5. Pendidikan Non-Formal

Praktik Kerja Lapangan di Departemen *Automation & Tooling* PT *TDK Electronics* Indonesia.

Sungail Liat, 25 Juli 2024

Tiara Yasmin Arimbi Chantika



LAMPIRAN 2

Program Mikrokontroler

PROGRAM ALAT

```
#include <Modbus.h>
#include <ModbusIP_ESP8266.h>
#include <WiFi.h>
#include <SPI.h>
#include <Ethernet2.h>
#include <DHT.h>
#include <Wire.h>
#define DHTPIN 14
#define DHTTYPE DHT22

float allPPM;
float averagePPM;
String quality = "";

//MQ-135
int MQ135_PIN = 33;
// Kalibrasi
float MQ135_RZERO = 76.63; // Resistansi sensor di udara bersih
float CO2_PARA = 116.6020682; // Parameter untuk kurva CO2
float CO2_PARB = 2.769034857;
float ppmco2 = 0.0;

//MQ-8
const int MQ8_PIN = 32;
// Kalibrasi
float H2_Ro = 10.0; // Nilai Ro untuk H2
float H2_Kc = 9.5; // Faktor konversi untuk H2

//MQ-2
const int MQ2_PIN = 34;
// Kalibrasi
float LPG_Ro = 10.0; // Nilai Ro untuk LPG
float LPG_Concentration_Ratio = 2.5;
float CH4_Ro = 9.0; // Nilai Ro untuk CH4
float CH4_Concentration_Ratio = 3.0;
```

```

//MQ-9
const int MQ9_PIN = 35;
// Kalibrasi
float MQ9_Ro = 10.0; // Nilai Ro dalam kondisi udara bersih
float MQ9_V_RL = 5.0; // Tegangan referensi RL (Resistor Load)
float MQ9_RL = 10.0; // Nilai RL (Resistor Load) dalam kilo-ohm
// Data titik kurva karakteristik sensor MQ-9
float CO_curve[3] = {2.3, 0.72, -0.34}; //(Data dari datasheet)

//DHT-22
DHT dht(DHTPIN, DHTTYPE);
float t = 0.0;
float h = 0.0;

//Set every sec read Sensor
unsigned long previousMillis = 0;
const long interval = 5000;

//Modbus
ModbusIP mb;

void setup() {
// put your setup code here, to run once:
dht.begin();
Serial.begin(9600); // Initialize serial communication at 115200 baud
analogReadResolution(12);

// Konfigurasi pin
pinMode(MQ2_PIN, INPUT);
pinMode(MQ8_PIN, INPUT);
pinMode(MQ9_PIN, INPUT);
pinMode(MQ135_PIN, INPUT);

WiFi.begin("Hehey Not Bad", "12345678");
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print("."); }

```



```

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

mb.server();
//Konfigurasi Holding Register Modbus
mb.addHreg(0); //for Temp int
mb.addHreg(1); //for Temp dec
mb.addHreg(2); //for Hum int
mb.addHreg(3); //for Hum dec
mb.addHreg(4); //for CO2 int
mb.addHreg(5); //for CO2 dec
mb.addHreg(6); //for CH4 int
mb.addHreg(7); //for CH4 dec
mb.addHreg(8); //for LPG int
mb.addHreg(9); //for LPG dec
mb.addHreg(10); //for H2 int
mb.addHreg(11); //for H2 dec
mb.addHreg(12); //for CO int
mb.addHreg(13); //for CO dec
mb.addHreg(14); //for MQ-2 value
mb.addHreg(15); //for MQ-8 value
mb.addHreg(16); //for MQ-9 value
mb.addHreg(17); } //for MQ-135 value

void loop() {
  delay(10);
  unsigned long currentMillis = millis(); // mendapatkan waktu sekarang
  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;

    // MQ2 Sensor
    int valueMQ2 = analogRead(MQ2_PIN); // Baca nilai MQ-2
    // Hitung nilai konsentrasi LPG dan CH4
    float LPG_Concentration = (float)valueMQ2 / LPG_Ro *
LPG_Concentration_Ratio;

```

```

    float CH4_Concentration = (float)valueMQ2 / CH4_Ro *
CH4_Concentration_Ratio;
    // Tampilkan nilai sensor di Serial Monitor
    Serial.print("Nilai sensor MQ-2 (analog): ");
    Serial.println(valueMQ2);
    Serial.print("Konsentrasi LPG: ");
    Serial.print(LPG_Concentration);
    Serial.println(" ppm");
    Serial.print("Konsentrasi CH4: ");
    Serial.print(CH4_Concentration);
    Serial.println(" ppm");
    // tampilkan nilai sensor pada Node-red
    int valuemq2_int = int (valueMQ2);
    mb.Hreg(14, valuemq2_int);
    int ppmch4_int = int (CH4_Concentration);
    int ppmch4_dec = int ((CH4_Concentration - ppmch4_int)*100);
    mb.Hreg(6, ppmch4_int);
    mb.Hreg(7, ppmch4_dec);
    int ppmlpg_int = int (LPG_Concentration);
    int ppmlpg_dec = int ((LPG_Concentration - ppmlpg_int)*100);
    mb.Hreg(8, ppmlpg_int);
    mb.Hreg(9, ppmlpg_dec);

// MQ8 Sensor
int valueH2 = analogRead(MQ8_PIN); // Baca nilai MQ-8
// Hitung konsentrasi gas H2
float ratioH2 = (float)valueH2 / H2_Ro;
float H2_Concentration = H2_Kc * pow(ratioH2, -1.3);
// Tampilkan nilai sensor di Serial Monitor
Serial.print("Nilai sensor MQ-8 (analog): ");
Serial.println(valueH2);
Serial.print("Konsentrasi H2: ");
Serial.print(H2_Concentration);
Serial.println(" ppm");
// tampilkan nilai sensor pada Node-red
int valuemq8_int = int (valueH2);
mb.Hreg(15, valuemq8_int);

```

```

int ppmh2_int = int (H2_Concentration);
int ppmh2_dec = int ((H2_Concentration - ppmh2_int)*100);
mb.Hreg(10, ppmh2_int);
mb.Hreg(11, ppmh2_dec);

// MQ9 Sensor
int valueMQ9 = analogRead(MQ9_PIN); // Baca nilai MQ-9
float MQ9_sensor_volt = ((float)valueMQ9 / 4095) * MQ9_V_RL;
float MQ9_RS_gas = ((MQ9_V_RL * MQ9_RL) / MQ9_sensor_volt) -
MQ9_RL; // Menghitung Resistansi sensor
// Hitung konsentrasi CO menggunakan kurva karakteristik
float ratioMQ9 = MQ9_RS_gas / MQ9_Ro; // Rasio Rs/Ro
float CO_concentration = pow(10, ((log(ratioMQ9) - CO_curve[1]) /
CO_curve[2]) + CO_curve[0]);
// Tampilkan nilai sensor di Serial Monitor
Serial.print("Nilai sensor MQ-9 (analog): ");
Serial.println(valueMQ9);
Serial.print("Konsentrasi CO: ");
Serial.print(CO_concentration);
Serial.println(" ppm");
// tampilkan nilai sensor pada Node-red
int valuemq9_int = int (valueMQ9);
mb.Hreg(16, valuemq9_int);
int ppmco_int = int (CO_concentration);
int ppmco_dec = int ((CO_concentration - ppmco_int)*100);
mb.Hreg(12, ppmco_int);
mb.Hreg(13, ppmco_dec);

// DHT22 Sensor
t = dht.readTemperature();
h = dht.readHumidity();
// Tampilkan nilai sensor di Serial Monitor
Serial.print("Humidity: ");
Serial.print(h);
Serial.print(" %, Temp: ");
Serial.print(t);
Serial.println(" Celsius");

```

```

// tampilkan nilai sensor pada Node-red
int t_int = int(t);
int t_dec = int((t - t_int) * 100);
mb.Hreg(0, t_int);
mb.Hreg(1, t_dec);
int h_int = int(h);
int h_dec = int((h - h_int) * 100);
mb.Hreg(2, h_int);
mb.Hreg(3, h_dec);

// MQ135 Sensor
int MQ135Value = analogRead(MQ135_PIN); // Membaca nilai MQ-135
float MQ135_voltage = MQ135Value * (3.3 / 4095.0);
float MQ135_RS_air = ((3.3 * 10.0) / MQ135_voltage) - 10.0;
float MQ135_ratio = MQ135_RS_air / MQ135_RZERO; // rasio Rs/Ro
ppmco2 = CO2_PARA * pow(MQ135_ratio, -CO2_PARB);
// Tampilkan nilai sensor di Serial Monitor
Serial.print("Nilai sensor MQ-135 (analog): ");
Serial.println(MQ135Value);
Serial.print("CO2 Concentration: ");
Serial.print(ppmco2);
Serial.println(" ppm");
// tampilkan nilai sensor pada Node-red
int valuemq135_int = int (MQ135Value);
mb.Hreg(17, valuemq135_int);
int ppmco2_int = int (ppmco2);
int ppmco2_dec = int ((ppmco2 - ppmco2_int)*100);
mb.Hreg(4, ppmco2_int);
mb.Hreg(5, ppmco2_dec);
// Menghitung Konsentrasi Gas di Udara
allPPM = valueMQ2 + valueH2 + valueMQ9 + MQ135Value;
averagePPM = allPPM / 4;
if ( averagePPM < 750 ){
    quality = " Baik "; }
else if ( averagePPM > 751 && averagePPM < 900 ){
    quality = " Sedang "; }
else if ( averagePPM > 901 && averagePPM < 1050 ){

```

```
    quality = " Tidak Sehat "; }  
else if ( averagePPM > 1051 && averagePPM < 1250 ){  
    quality = " Sangat Tidak Sehat "; }  
else if ( averagePPM > 1251 ) {  
    quality = " Berbahaya "; }  
Serial.println(averagePPM);  
Serial.print("Air Quality = ");  
Serial.println(quality);  
//Modbus Write  
mb.task();  
}
```





LAMPIRAN 3

Flow Node-RED

Flow Node-RED

