**LAPORAN TUGAS AKHIR**

**SISTEM PEMANTAUAN TINGKAT POLUSI UDARA PADA LINGKUNGAN TERMINAL BUS DENGAN MENGGUNAKAN MEDIA THINGSPEAK**

**Logo

Description automatically generated**

Oleh :

Gusti Riyan Ramadana

5191011017

**PROGRAM STUDI S-1 TEKNIK KOMPUTER**

**FAKULTAS SAINS & TEKNOLOGI**

**UNIVERSITAS TEKNOLOGI YOGYAKARTA**

**YOGYAKARTA**

**2024**

# LEMBAR PERNYATAAN

Yang bertanda tangan dibawah ini

Nama : Gusti Riyan Ramadana

NPM : 5191011017

Program Studi : Teknik Komputer

Program : Sarjana

Fakultas : Sains dan Teknologi

Menyatakan bahwa laporan tugas akhir dengan judul Sistem Pemantauan Tingkat Polusi Udara Pada Lingkungan Terminal Bus Dengan Menggunakan Media Thingspeak adalah hasil pekerjaan saya sendiri dan tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan disuatu perguruan tinggi. Sepanjang sepengetahuan saya, laporan tugas akhir ini tidak terdapat karya yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dicantumkan dalam naskah ini dengan mengikuti tata cara dan etika penulisan karya ilmiah yang lazim. Apabila ternyata terbukti bahwa pernyataan ini tidak benar sepenuhnya, menjadi tanggung jawab saya.

Yogyakarta,

Gusti Riyan Ramadana

# KATA PENGANTAR

Puji syukur kami panjatkan kehadirat Tuhan Yang Maha Esa atas segala Rahmat, Taufiq, Hidayah, dan Inayah-Nya. Hanya karena kekuatan dan bimbingan-Nya, kami dapat menyelesaikan laporan tugas akhir di. kami menyadari masih banyak hal yang perlu digali lebih lanjut, terutama mengenai hal-hal baru yang kami temui pada penelitian ini. Kami berkomitmen untuk terus mengembangkan diri sehingga kami dapat menjadi peneliti yang berkualitas. Dengan rasa hormat saya mengucapkan terimakasih sebesar-besarnya kepada:

1. Dr. Bambang Martono Setiawan, M.M., C.A., Akt. selaku Rektor Universitas Teknologi Yogyakarta
2. Sutarman, M. Kom., Ph.D. selaku Dekan Fakultas Sains dan Teknologi (FST).
3. Adam Sekti Aji, S. Kom, M. Kom. selaku ketua Program Studi Teknik Komputer.
4. Rodhiyah Mardhiyyah, S.Kom., M.Kom. selaku dosen pembimbing yang telah meluangkan waktu untuk mengarahkan penulis dalam menyelesaikan laporan Kerja Praktik.
5. Orang tua dan keluarga penulis yang telah banyak memberikan bantuan dukungan moril maupun materil.

Penulis

Gusti Riyan Ramadana

# DAFTAR ISI

[LEMBAR PERNYATAAN i](#_Toc167848978)

[KATA PENGANTAR ii](#_Toc167848979)

[DAFTAR ISI iii](#_Toc167848980)

[DAFTAR GAMBAR v](#_Toc167848981)

[DAFTAR TABEL vi](#_Toc167848982)

[ABSTRAK vii](#_Toc167848983)

[ABSTRACT viii](#_Toc167848984)

[BAB I PENDAHULUAN 1](#_Toc167848985)

[1.1 Latar belakang 1](#_Toc167848986)

[1.2 Rumusan Masalah 2](#_Toc167848987)

[1.3 Batasan Masalah 2](#_Toc167848988)

[1.4 Tujuan Penelitian 3](#_Toc167848989)

[1.5 Manfaat Penelitian 3](#_Toc167848990)

[1.6 Sistematika Penulisan 3](#_Toc167848991)

[BAB II TINJAUAN PUSTAKA 5](#_Toc167848992)

[2.1 Kajian Hasil Penelitian 5](#_Toc167848993)

[2.2 Landasan Teori 8](#_Toc167848994)

[2.2.1 Polusi Udara 8](#_Toc167848995)

[2.2.2 Sensor Polusi Udara 9](#_Toc167848996)

[2.2.3 Mikrokontroler 10](#_Toc167848997)

[2.2.4 Kipas Polusi Udara 11](#_Toc167848998)

[2.2.5 Internet of Things 12](#_Toc167848999)

[2.2.6 ThingSpeak 12](#_Toc167849000)

[BAB III METOFOLOGI PENELITIAN 14](#_Toc167849001)

[3.1 Langkah Penelitian 14](#_Toc167849002)

[3.1.1 Analisis Masalah Sumber Polusi Udara 15](#_Toc167849003)

[3.1.2 Analisis Kebutuhan Alat dan Bahan 15](#_Toc167849004)

[3.1.3 Membuat Desain Rancangan dan Simulasi 16](#_Toc167849005)

[3.1.4 Implementasi dan Pengujian 16](#_Toc167849006)

[3.2 Alat dan Bahan 17](#_Toc167849007)

[3.2.1 Software 17](#_Toc167849008)

[3.2.2 Hardware 19](#_Toc167849009)

[3.2.3 Komponen Elektronika dan Mikrokontroler 20](#_Toc167849010)

[3.3 Rencana Pengujian 26](#_Toc167849011)

[3.3.1 Pengujian Sensor 27](#_Toc167849012)

[3.3.2 Pengujian Kipas Polusi Udara 27](#_Toc167849013)

[3.3.3 Pengujian Antarmuka Pengguna 27](#_Toc167849014)

[BAB IV ANALISIS DAN PERANCANGAN SISTEM 28](#_Toc167849015)

[4.1 Rencana Pengembangan Sistem 28](#_Toc167849016)

[4.1.1 Blok Diagram Sistem 28](#_Toc167849017)

[4.1.2 Flowchart Sistem 29](#_Toc167849018)

[4.2 Perancangan Sistem 30](#_Toc167849019)

[4.2.1 Rangkaian Sensor MQ-135 30](#_Toc167849020)

[4.2.2 Rangkaian Sensor GP2Y1014AU0F 31](#_Toc167849021)

[4.2.3 Rangkaian Fan DC 31](#_Toc167849022)

[4.2.3 Rancangan ThingSpeak 32](#_Toc167849023)

[BAB V HASIL DAN PEMBAHASAN 34](#_Toc167849024)

[5.1 Impelementasi 34](#_Toc167849025)

[5.1.1 Implementasi sensor MQ-135 34](#_Toc167849026)

[5.1.2 Implementasi Sensor GP2Y1014AU0F 36](#_Toc167849027)

[5.1.3 Implementasi Fan DC 38](#_Toc167849028)

[5.1.4 Implementasi ThingSpeak 40](#_Toc167849029)

[5.2 Pengujian Sistem 44](#_Toc167849030)

[5.2.1 Pengujian Sensor 44](#_Toc167849031)

[5.2.2 Pengujian Fan DC 48](#_Toc167849032)

[5.2.3 Pengujian ThingSpeak 49](#_Toc167849033)

[BAB VI PENUTUP 52](#_Toc167849034)

[6.1 Kesimpulan 52](#_Toc167849035)

[6.2 Saran 52](#_Toc167849036)

# DAFTAR GAMBAR

[Gambar 3. 1 Flowchart Langkah Penelitian 14](#_Toc167770251)

[Gambar 3. 2 Diagram Blok Langkah Penelitian 15](#_Toc167770252)

[Gambar 3. 3 Software Arduino IDE 18](#_Toc167770253)

[Gambar 3. 4 Tampilan Platform ThingSpeak 19](#_Toc167770254)

[Gambar 3. 5 ESP32 21](#_Toc167770255)

[Gambar 3. 6 Sensor MQ-135 22](#_Toc167770256)

[Gambar 3. 7 Sensor GP2Y1014AU0F 23](#_Toc167770257)

[Gambar 3. 8 Fan DC 24](#_Toc167770258)

[Gambar 3. 9 Driver Motor L298N 25](#_Toc167770259)

[Gambar 3. 10 Kabel Jumper 26](#_Toc167770260)

[Gambar 4. 1 Blok Diagram Sistem 28](#_Toc167770241)

[Gambar 4. 2 Flowchart Sistem 29](#_Toc167770242)

[Gambar 4. 3 Rangkaian Sensor MQ-135 30](#_Toc167770243)

[Gambar 4. 4 Rangkaian Sensor GP2Y1014AU0F 31](#_Toc167770244)

[Gambar 4. 5 Rangkaian Fan DC 32](#_Toc167770245)

[Gambar 4. 6 Rancangan Layout ThingSpeak 33](#_Toc167770246)

[Gambar 5. 1 Implementasi Sensor MQ-135 34](#_Toc167770223)

[Gambar 5. 2 Kode Program Sensor MQ-135 35](#_Toc167770224)

[Gambar 5. 3 Implementasi Sensor GP2Y1014AU0F 36](#_Toc167770225)

[Gambar 5. 4 Kode Program Sensor GP2Y1014AU0F 37](#_Toc167770226)

[Gambar 5. 5 Implementasi Fan DC 39](#_Toc167770227)

[Gambar 5. 6 Konfigurasi Pin Fan DC 39](#_Toc167770228)

[Gambar 5. 7 Kode Program Mengaktifkan Fan DC 40](#_Toc167770229)

[Gambar 5. 8 Pembuatan Channel ThingSpeak 41](#_Toc167770230)

[Gambar 5. 9 Konfigurasi Awal ThingSPeak 41](#_Toc167770231)

[Gambar 5. 10 Kode Proram Mengaktifkan ThingSpeak 43](#_Toc167770232)

[Gambar 5. 11 Kode Program Field ThingSpeak 43](#_Toc167770233)

[Gambar 5. 12 Serial Monitor Pengujian Sensor MQ-135 45](#_Toc167770234)

[Gambar 5. 13 Serial Monitor Sensor GP2Y1014AU0F 47](#_Toc167770235)

[Gambar 5. 14 Grafik Hasil Pengujian ThingSpeak 49](#_Toc167770236)

# DAFTAR TABEL

[Tabel 2. 1 Tabel Perbandingan Penelitian 6](#_Toc167770326)

[Tabel 3. 1 Spesifikasi Laptop 20](#_Toc167770312)

[Tabel 3. 2 Spesifikasi SmartPhone 20](#_Toc167770313)

[Tabel 3. 3 Spesifikasi ESP32 22](#_Toc167770314)

[Tabel 3. 4 Spesifikasi Sensor MQ-135 23](#_Toc167770315)

[Tabel 3. 5 Spesifikasi Sensor GP2Y1014AU0F 24](#_Toc167770316)

[Tabel 3. 6 Spesifikasi Fan DC 25](#_Toc167770317)

[Tabel 3. 7 Spesifikasi Driver Motor 26](#_Toc167770318)

[Tabel 5. 1 Pengujian Sensor MQ-135 45](#_Toc168585581)

[Tabel 5. 2 Pengujian Jarak Sensor MQ-135 46](#_Toc168585582)

[Tabel 5. 3 Pengujian Sensor GP2Y1014AU0F 47](#_Toc168585583)

[Tabel 5. 4 Pengujian Fan DC 48](#_Toc168585584)

[Tabel 5. 5 Pengujian ThingSpeak 49](#_Toc168585585)

[Tabel 5. 6 Pengujian Keakuratan Grafik ThingSpeak 50](#_Toc168585586)

# ABSTRAK

Pencemaran udara telah menjadi perhatian global, termasuk di terminal bus, karena dampak seriusnya terhadap kesehatan manusia dan lingkungan. Terminal bus di Yogyakarta merupakan salah satu pusat transportasi darat yang berperan penting dalam mobilitas pengunjung dan pekerja di sekitar terminal. Namun, keberadaan terminal bus juga berdampak masalah polusi udara pada lalu lintas sekitarnya. Sebagai solusi, alat monitoring polusi udara berbasis Internet of Things (IoT) dengan kipas polusi udara diusulkan. Alat ini Dilengkapi dengan sensor MQ-135 dan GP2Y1014AU0F yang dapat mendeteksi zat menyebabkan pencemaran udara dan mengirimkan data secara real-time ke server ThingSpeak melalui koneksi IoT. Sensor gas MQ-135 ini memiliki dapat mendetksi zat polusi CO. Dengan menggunakan modul mikrokontroler, sensor dapat diintegerasikan untuk pemantauan secara real-time melalui platform ThingSpeak dan mengatur ambang batas berbahaya untuk mengaktifkan Fan DC untuk menetralisir udara. Sensor GP2Y1014AU0F mendeteksi partikel-partikel debu kecil dalam udara yang dapat berdampak negatif pada kesehatan manusia jika terhirup dalam jumlah besar. Dengan memanfaatkan sensor GP2Y1014AU0F yang terhubung ke mikrokontroler, dapat diintegrasikan untuk pemantauan secara real-time melalui platform ThingSpeak dan dapat mengatur ambang batas untuk mengaktifkan fan DC sebagai kipas polusi udara. Mikrokontroler berperan sebagai otak sistem, yang dapat diprogram untuk mengontrol fan DC sesuai dengan tingkat polusi udara yang diukur oleh sensor yang sesuai. Ketika MQ-135 dan GP2Y1014AU0F mendeteksi kadar CO melebihi 8 PPM dan partikel debu melebihi 9 µg/m³, mikrokontroler akan memberikan instruksi kepada fan DC untuk meningkatkan sirkulasi udara yang dapat menetralisir udara dalam waktu 1 sampa 2 menit. ThingSpeak pada penelitian ini digunakan sebagai aplikasi pemantauan kadar polusi yang dideteksi oleh sensor MQ-135 dan GP2Y1014AU0F yaitu karbon monoksida (CO) dan partikel debu (PM). platform ThingSpeak sebagai media pemantauan yang menampilkan grafik secara real-time sehingga dapat melalakukan deteksi dini tingkat polusi udara. Sehingga tidak hanya dapat memantau tingkat polusi sistem ini juga dapat menetralisir tingkat polusi di sekitar terminal.

Kata Kunci: Fan DC, GP2Y1014AU0F, MQ-135, Polusi Udara, ThingSpeak

# ABSTRACT

Air pollution has become a global concern, including at bus stations, because of its serious impact on human health and the environment. The bus station in Yogyakarta is a land transportation center that plays an important role in the mobility of visitors and workers around the bus station. However, the existence of the bus station also has an impact on air pollution problems in the surrounding traffic. As a solution, an Internet of Things (IoT) based air pollution monitoring tool with an air pollution fan is proposed. This tool is equipped with MQ-135 and GP2Y1014AU0F sensors which can detect substances causing air pollution and send real-time data to the ThingSpeak server via an IoT connection. This MQ-135 gas sensor can detect CO pollution. Using a microcontroller module, sensors can be integrated for real-time monitoring via the ThingSpeak platform and set dangerous thresholds to activate DC Fans to neutralize air. The GP2Y1014AU0F sensor detects small dust particles in the air which can have a negative impact on human health if inhaled in large quantities. By utilizing the GP2Y1014AU0F sensor connected to a microcontroller, it can be integrated for real-time monitoring via the ThingSpeak platform and can set the threshold for activating the DC fan as an air pollution fan. The microcontroller acts as the brain of the system, which can be programmed to control the DC fan according to the level of air pollution measured by the appropriate sensors. When the MQ-135 and GP2Y1014AU0F detect CO levels exceeding 8 PPM and dust particles exceeding 9 µg/m³, the microcontroller will give instructions to the DC fan to increase air circulation which can neutralize the air within 1 to 2 minutes. ThingSpeak in this research is used as an application to monitor pollution levels detected by the MQ-135 and GP2Y1014AU0F sensors, namely carbon monoxide (CO) and dust particles (PM). The ThingSpeak platform is a monitoring medium that displays real-time graphics so that it can carry out early detection of air pollution levels. So not only can this system monitor pollution levels, it can also neutralize pollution levels around the bus station.

Keywords: Air Pollution, DC Fan, GP2Y1014AU0F, MQ-135, ThingSpeak

# BAB I

**PENDAHULUAN**

## 1.1 Latar belakang

Pencemaran udara telah menjadi perhatian global, termasuk di terminal bus, karena dampak seriusnya terhadap kesehatan manusia dan lingkungan. Sistem pemantauan konvensional yang umumnya mahal dan kurang akurat telah menyulitkan respons cepat terhadap perubahan polusi udara. Terminal bus di Yogyakarta merupakan salah satu pusat transportasi darat yang berperan penting dalam mobilitas penduduk, pengunjung dan pekerja di sekitar terminal. Namun, keberadaan terminal bus juga berdampak masalah polusi udara pada lalu lintas di sekitarnya. Sektor transportasi merupakan penyumbang utama polusi udara di Daerah Istimewa Yogyakarta. Emisi gas buang yang ditimbulkan Bus dan Travel minibus berdasarkan skenario bussiness as usual adalah Carbon Dioxide NonBiogenic sebesar 212.100 metrik ton, Carbon Monoxide sebesar 1400 metrik ton, Nitrogen Oxides sebesar 2900 metrik ton, NonMethane Volatile Organic Compounds sebesar 500 metrik ton, dan methane dan nitrous oxide yang nilainya sangat kecil di bawah 200 metrik ton (Zulem et al., (2020). peningkatan pencemaran udara sebanyak 70% terutama disebabkan oleh kadar partikel debu halus. Oleh karena itu, diperlukan solusi untuk pemantauan polusi udara dan meningkatkan kesadaran tentang pentingnya menjaga kualitas udara.

Masalah utama melibatkan keterbatasan sistem pemantauan yang ada tidak mampu memberikan penanganan terhadap tingkat pencemaran yang tinggi. Situasi tersebut menyebabkan orang yang bekerja di sekitar terminal rentan terhadap serangan penyakit yang diakibatkan pencemaran udara. Dengan adanya sistem pemantauan secara langsung, memungkinkan orang yang bekerja di sekitar terminal dapat memantau tingkat polusi udara serta melakukan tindakan cepat untuk mengurangi dampak buruk dari polusi yang tinggi.

Sebagai solusi, alat monitoring polusi udara berbasis Internet of Things (IoT) dengan kipas polusi udara diusulkan. Alat ini Dilengkapi dengan sensor MQ-135 dan GP2Y1014AU0F yang dapat mendeteksi zat karbon monoksida (CO) dan partikel debu yang menyebabkan pencemaran udara dan mengirimkan data secara real-time ke server melalui koneksi IoT. Alat ini dilengkapi kipas DC yang berfungsi sebagai penggerak kipas agar dapat menghebuskan angin sehingga polusi tidak masuk ke area yang padat orang seperti kantin, ruang tunggu dan lainnya. Kipas ini dapat membantu untuk mengurangi tingkat polusi udara.

Cara kerja alat ini adalah ketika polusi udara yang diakibatkan kendaraan masuk ke area yang padat orang seperti kantin dan ruang tunggu maka sensor akan mendeteksi dan menampilkan tingkat polusi kedalam aplikasi berbasis IoT yaitu ThingSpeak yang dapat dipantau melalui aplikasi ThingView oleh orang yang bekerja di area terminal. Ketika kadar CO melebihi 8 PPM dan partikel debu melebihi 9 µg/m³maka kipas DC akan aktif untuk menghembuskan angin agar polusi keluar dari area yang padat orang. Sehingga tidak hanya dapat memantau tingkat polusi sistem ini juga dapat menetralisir tingkat polusi di sekitar terminal.

## 1.2 Rumusan Masalah

Berdasarkan hasil penelitian beberapa masalah yang harus dipecahkan oleh peneliti:

1. Apakah sistem monitoring polusi udara berbasis Internet of Things mampu melakukan deteksi dini terkait tingkat polusi udara?
2. Apakah sistem monitoring polusi udara berbasis Internet of Things ini dapat mengurangi tingkat polusi udara?

## 1.3 Batasan Masalah

Adapun batasan masalah dalam sistem monitoring polusi udara berbasis Internet of Things (IoT):

1. Alat ini difokuskan untuk mendeteksi polusi udara dan kipas untuk mengurangi tingkat polusi udara dengan pematauan melalui thingspeak yang bisa dipantau melalui aplikasi thingview.
2. Difokuskan untuk mendeteksi kadar polusi CO dan partikel debu.
3. Sensor yang digunakan yaitu sensor MQ-135 untuk mendeteksi CO dan sensor GP2Y1014AU0F untuk mendeteksi partikel debu

## 1.4 Tujuan Penelitian

Adapun tujuan dari pembuatan sistem monitoring polusi udara berbasis Internet of Things (IoT):

1. Mengimplementasikan sistem monitoring polusi udara berbasis Internet of Things (IoT) yang dapat digunakan untuk memantau dan mengatasi masalah polusi udara
2. Memantau Tingkat polusi udara yang dihasilkan oleh asap kendaraan di terminal bus
3. Mengurangi tingkat polusi udara yang dihasilkan oleh asap kendaraan di terminal bus

## 1.5 Manfaat Penelitian

Adapun manfaat dari pembuatan pendeteksi polusi udara berbasis Internet of Things (IoT) sebagai berikut:

1. Sebagai bentuk kontribusi membantu perkembangan tekonologi berbasis Internet of Things (IoT)
2. Membantu memantau dan mengurangi tingkat polusi udara serta menjaga Kesehatan lingkungan

## 1.6 Sistematika Penulisan

Sistematika yang digunakan dalam penyusunan laporan tugas akhir ini adalah sebagai berikut:

**BAB I PENDAHULUAN**

Pada bab ini berisikan latar belakang penelitian, rumusan masalah, Batasan masalah, tujuan penelitian, manfaat penelitian serta sistematika penulisan yang digunakan pada laporan untuk dasar dalam sistem monitoring polusi udara berbasis Internet of Things

**BAB II TINJAUAN PUSTAKA**

Pada bab ini mempelajari tentang penelitian terdahulu dalam membuat atau rancangan yang telah dilakukan peneliti terdahulu untuk mendapatkan wawasan secara mendalam mengenai permasalahan*.*

**BAB III METODOLOGI PENELITIAN**

Pada bab ini menjelasakan langkah penelitian serta alat dan bahan yang digunakan dalam sistem monitoring polusi udara berbasis Internet of Things (IoT).

**BAB IV ANALISIS DAN PERANCANGAN SISTEM**

Pada bab ini berisikan tentang seluruh rencana rancangan yang akan dilakukan dalam penelitian. Mulai dari program, penyesuaian alat dan bahan dan rancangan sistem monitoring polusi udara berbasis Internet of Things.

**BAB V HASIL DAN PEMBAHASAN**

Pada bab ini menjelaskan hasil dan pembahasan dari sistem monitoring polusi udara berbasis Internet of Things.

**BAB VI PENUTUP**

Pada bab ini menjelaskan kesimpulan dan saran dari Hasil pembuatan sistem monitoring polusi udara berbasis Internet of Things.

# BAB II

**TINJAUAN PUSTAKA**

## 2.1 Kajian Hasil Penelitian

Hasil penelitian (Hakim & Susanto, (2020) dalam jurnal yang berjudul Sistem Monitoring Kualitas Udara Berbasis Internet of Things, Sistem monitoring polusi udara berbasis IoT merupakan teknologi yang digunakan untuk mengukur dan mengatur kualitas udara. Dengan menggunakan teknologi IoT, sistem ini mampu mengumpulkan dan mengirim data secara real-time tentang kualitas udara ke platform data online. Ini memungkinkan pengguna untuk memantau dan mengatur kualitas udara secara otomatis melalui internet. Sistem ini menggunakan berbagai sensor, seperti sensor suhu, kelembaban, dan pembacaan gas, seperti CO, NO, dan O3. Data dari sensor ini akan kemudian dikonversi menggunakan teori hukum ohm dan dikirim ke platform data online melalui teknologi GPRS. Dengan menggunakan teknologi IoT, sistem ini dapat memantau kualitas udara di berbagai lokasi dan membantu mengendalikan polusi udara.

Hasil penelitian (Fachrizal et al., (2022) dalam jurnal yang berjudul Sistem Monitoring Polusi Udara Mengggunakan Sensor Nitrogen Carbon Berbasis Internet of Things, Cara kerja sistem monitoring polusi udara berbasis IoT (Internet of Things) dapat diterangkan melalui beberapa contoh dan studi yang telah dilakukan. Beberapa contoh sistem ini melibatkan penggunaan sensor-sensor yang berbeda untuk mendeteksi gas polutan, seperti sensor Nitrogen Carbon (CO, NO2 dan NH3) sensor gas karbon monoksida (CO) dan karbon dioksida (CO2) sensor MQ137 untuk pendeteksi amonia (NH3) Sistem ini menggunakan mikrokontroler seperti Arduino Uno, ESP32, atau NodeMCU untuk mengumpulkan data dari sensor dan mengirimkannya ke cloud server melalui teknologi LoRA atau ESP32. Data tersebut kemudian dapat ditampilkan melalui web page atau aplikasi pada smartphone atau komputer PC.

Hasil penelitian (Susilo & David, (2023) dalam jurnal yang berjudul Sistem Pemantauan Gas Berbahaya Pada Peternakan Ayam Berbasis Internet of Things, Dalam sistem monitoring polusi udara berbasis IoT, perangkat yang digunakan berbeda berdasarkan tujuan dan kondisi lingkungan yang diukur. Contohnya, dalam penelitian tentang pemantauan gas berbahaya pada peternakan ayam, perangkat yang digunakan adalah sensor MQ-4 untuk deteksi metana, sensor MQ-135 untuk deteksi amonia, dan ESP8266 sebagai microcontroller untuk perangkat IoT. Hasil penelitian (Aghorru & Koprawi, (2023) dalam jurnal yang berjudul Rancang Bangun Sistem Pemantau Kualitas Dan Polusi Udara PM2.5 Yang Terintegrasi Dengan Platform IoT, Dalam pengembangan sistem pemantauan kualitas dan polusi udara PM2.5 yang terintegrasi dengan platform IoT, perangkat yang digunakan meliputi sensor debu PM2.5 GP2Y1010AU0F, sensor suhu dan kelembapan DHT22, dan mikrokontroler nodemcu dengan modul Wifi esp8266.

Hasil penelitian (Nur Kurnia Fibiani & Rini puji astutik, (2023) dalam jurnal yang berjudul Sistem Monitoring Kosentrasi Amonia (Nh3) Pada Kawasan Pt. Petrokimia Gresik Berbasis IoT, Penelitian sistem monitoring polusi udara berbasis IoT menunjukkan bahwa sistem ini dapat digunakan untuk mengukur dan mengelola kualitas udara di berbagai kawasan, termasuk kawasan industri. Sistem ini menggunakan sensor MQ137 untuk deteksi kosentrasi gas ammonia (NH3) dan sensor GP2Y1014AU0F untuk deteksi debu hasil dari sensor tersebut akan diproses melalui ESP32 dan akan ditampilkan melalui panel LED P5 sebagai pusat informasi dari hasil pengambilan data.

Tabel 2. 1 Tabel Perbandingan Penelitian

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Judul | Penulis | tahun | Hasil penelitian |
| 1 | Sistem Monitoring Kualitas Udara Berbasis Internet of Things | Hakim & Susanto | 2020 | Sistem monitoring polusi udara terealisasi dengan menggunakan sensor MQ-135, MQ-131 dan DHT22 untuk mengukur parameter CO, NO, O3, suhu, dan kelembapan serta integrasi sistem dengan SIM900A sebagai alat komunikasi data dan ThingSpeak untuk menampilkan data. |
| 2 | Sistem Monitoring Polusi Udara Mengggunakan Sensor Nitrogen Carbon Berbasis Internet of Things | Ferry Fachrizal, Julham & Antoni | 2022 | Penggunaan Modul sensor MICS 6814 dapat digunakan untuk mendeteksi terjadinya polusi pada udara dengan memperhitungkan kandungan NH3, CO dan NO2 dan membandingkan nilai hasil sensor dengan index polutan udara serta Untuk mengirim data dari modul sensor digunakan sistem komunikasi LoRA yang terdiri dari LoRA Shield dan LoRA gateway. |
| 3 | Sistem Pemantauan Gas Berbahaya Pada Peternakan Ayam Berbasis Internet of Things | Susilo & David | 2023 | Penelitian ini mengusulkan pemantauan gas berbahaya tmenggunakan sensor dan berbasis Internet of Things (IoT) dan peternak dapat melakukan pemantauan dari jarak jauh dengan menggunakan perangkat smart phone. Perangkat yang digunakan dalam penelitian ini adalah sensor MQ-4 untuk deteksi metana, sensor MQ-135 untuk deteksi amonia dan ESP8266 sebagai microcontroller untuk perangkat IoT. Hasil penelitian ini dapat membantu peternak untuk menjaga kualitas udara di lingkungan kandang peternakan ayam yang dikelolanya. |
| 4 | Rancang Bangun Sistem Pemantau Kualitas Dan Polusi Udara PM2.5 Yang Terintegrasi Dengan Platform IoT | Aghorru & Koprawi | 2023 | Alat ini menggunakan dua sensor utama, yaitu sensor debu PM2.5 GP2Y1010AU0F dan sensor suhu dan kelembapan DHT22. Semua sensor dihubungkan ke esp8266. Data dari sensor ditampilkan secara real-time yang dipantau melalui smartphone atau perangkat koneksi internet serta dapat mengirimkan notifikasi tentang kondisi udara di sekitar sesuai dengan nilai ambang batas PM2.5 BMKG. Pengujian alat menunjukkan bahwa sensor PM2.5 yang dirancang memiliki standar deviasi sebesar 22,1 dan rata-rata sebesar 47,0. |
| 5 | Sistem Monitoring Kosentrasi Amonia (Nh3) Pada Kawasan Pt. Petrokimia Gresik Berbasis IoT | Nur Kurnia Fibiani & Rini puji astutik | 2023 | Dengan menggunakan sensor MQ137 sebagai pendeteksi gas ammonia (NH3) dan sensor GP2Y1014AU0F (sensor pendeteksi debu). Hasil dari sensor tersebut akan diproses melalui ESP32 sebagai (unit process) dan akan ditampilkan melalui panel LED P5 sebagai pusat informasi dari hasil pengambilan data. Jika kosentrasi gas Ammonia lebih dari 200 ppm, maka area kawasan tersebut terindikasi sebagai kawasan dengan kadar ammonia cukup tinggi. |

Tabel 2.1 adalah perbandingan dari lima penelitian tentang sistem monitoring polusi udara berbasis Internet of Things dari berbagai macam pendekatan. Dalam penelitan terdapat berbagai macam sensor gas dan sensor partikel udara yang digunakan, seperti MQ-135, MQ-131, MICS 6814, MQ-4 dan GP2Y1010AU0F, untuk mengukur parameter seperti CO, NO, O3, amonia (NH3), dan PM2.5. ESP8266 dan ESP32 digunakan untuk mengintegrasikan data dari sensor-sensor ini ke dalam sistem IoT. Selain itu, komunikasi data menggunakan berbagai platform, seperti SIM900A, LoRA, dan penggunaan panel LED P5 untuk menampilkan informasi.

## 2.2 Landasan Teori

Sistem monitoring polusi udara berbasis Internet of Things (IoT) adalah sebuah sistem yang memanfaatkan teknologi sensor, jaringan komunikasi, dan komputasi untuk memantau dan mengukur kualitas udara. Dalam merancang pendeteksi polusi udara berbasis IoT, ada beberapa landasan teori yang perlu dipertimbangkan:

### 2.2.1 Polusi Udara

Polusi udara merupakan masalah lingkungan yang berbahaya bagi kesehatan manusia. Dampaknya lebih parah dalam ruangan ketimbang di luar ruangan. Faktor-faktor yang mengakibatkan polusi udara termasuk penggunaan bahan bakar memasak, kendaraan bermotor yang menggunakan bahan bakar minyak bumi, dan industri Polusi udara dapat menyebabkan penyakit pneumonia atau paru-paru basah (A'yun et al., (2023).

Polusi debu dapat mengganggu kesehatan pekerja, seperti yang terjadi di perusahaan CV. Accent House Pecangaan Jepara. Polusi juga dapat menyebabkan penyakit saluran pernapasan akut (ISPA) pada balita. Untuk meminimalisasi bahaya polusi, perlu dilakukan kegiatan pengendalian polusi, seperti pengumpulan data dan pengendalian kadar debu di lingkungan kerja (Mauluddin et al., (2023).

Polusi tinggi juga biasa muncul di terminal bus. Terutama berasal dari emisi kendaraan bermotor yang terjadi di jalan raya dan dapat mengakibatkan gangguan fungsi paru pada sopir bus. Kebiasaan merokok juga dapat mempengaruhi fungsi paru, sehingga pengaruhnya harus diperhatikan (Suharti & Br. Surbakti, 2020). Dalam sebuah studi yang diterbitkan di jurnal Environmental Science and Pollution Research, dinyatakan bahwa paparan CO di atas 9 ppm dapat meningkatkan risiko gangguan kardiovaskular dan pernapasan (Bhalli et al., 2016). Batas konsentrasi debu yang dapat terhirup yang direkomendasikan oleh American Conference of Governmental Industrial Hygienists (ACGIH) adalah 3 mg/m3 untuk paparan waktu 8 jam (Yassin et al., 2005).

### 2.2.2 Sensor Polusi Udara

Sensor polusi udara adalah alat yang digunakan untuk mendeteksi dan mengukur kualitas udara di lingkungan. Dengan teknologi IoT, data dari sensor dapat dimonitor dan dipantau secara real-time melalui internet. Sensor polusi udara IoT dapat mengukur kadar polusi seperti karbon monoksida, debu PM2.5, dan gas lainnya yang dapat mengancam kesehatan manusia. Alat ini berguna untuk mengidentifikasi dan mengukur tingkat polusi udara di sekitar kampus, ruangan, dan lingkungan lainnya. Dengan data yang diperoleh, dapat dilakukan tindakan mitigasi dan pencegahan untuk meningkatkan kualitas udara (Fikri et al., (2023).

Cara kerja sensor polusi udara tergantung pada tipe dan desain alat pengukur. Sensor MQ-135 digunakan untuk mendeteksi gas dan menghasilkan nilai tegangan yang berhubungan dengan kadar gas. sensor ini menggunakan teknologi MOSFET untuk mendeteksi gas dan memiliki kemampuan untuk mendeteksi gas karbon monoksida sampai dengan kadar 50 ppm dan gas karbon dioksida sampai dengan kadar 1000 ppm (Humairoh & Putra, (2021). Sensor ini digunakan untuk mendeteksi partikel udara berukuran 2.5 µm dan menghasilkan nilai tegangan yang berhubungan dengan kadar partikel (Salamah et al., (2022).

Sensor MQ-135 dan sensor GP2Y1014AU0F memiliki beberapa kelemahan. Sensor MQ-135 memiliki kekurangan yaitu Kesulitan dalam deteksi gas karbon monoksida (CO) yang tinggi (Pendriadi et al., (2023). Selain itu, Kekurangan sensor GP2Y1014AU0F adalah tidak mampu mendeteksi partikel lebih kecil dari 0,3 μm (Crnosija et al., (2022).

### 2.2.3 Mikrokontroler

Mikrokontroler adalah sebuah perangkat lunak atau perangkat elektronik yang mengontrol dan mengatur sistem elektronik yang lebih kompleks. Mikrokontroler mampu menangani berbagai fungsi, seperti pengaturan sensor, pengaturan motor, dan pengaturan sistem komunikasi. Mikrokontroler ini sering digunakan dalam berbagai aplikasi, seperti pengaturan sistem parkir, pengukuran tekanan udara, dan pengendalian robot. Mikrokontroler mampu menerima input dan mengirimkan output ke perangkat keras lainnya, seperti motor, sensor, dan display (Budihartono & Afriliana, (2019).

Terdapat beberapa jenis mikrokontroler yang digunakan dalam Implementasi alat IoT. ESP32 diproduksi oleh Espressif System dan menggunakan firmware dengan menggunakan bahasa pemrograman (Gito Resmi et al., (2023). Setelah itu ada wemos D1 memiliki keunggulan dari segi spesifikasi dimana, wemos D1 memiliki core yakni ESP32 yang mempunyai prosesor 32-bit (Hamid et al., (2023). Arduino memiliki prosesor Atmel AVR dan softwarenya memiliki bahasa pemrograman sendiri dengan menggunakan Arduino IDE (Ramli & Arief, (2021).

Mikrokontroler ESP32 dapat digunakan dalam perancangan sistem monitoring polusi udara berbasis IoT. Misalnya, dalam perancangan sistem monitoring kecepatan angin dan temperatur udara berbasis IoT, mikrokontroler ESP32 digunakan menggunakan sensor anemometer, BMP 280, dan aplikasi Blynk untuk mempermudah dalam proses monitoring. Sistem ini mampu memonitoring kecepatan angin dan temperatur udara dari jarak jauh oleh pengguna menggunakan aplikasi Blynk (Manalu & Gunoto, (2023).

### 2.2.4 Kipas Polusi Udara

Kipas dalam pendeteksi polusi udara berbasis IoT adalah komponen yang dapat mengurangi polusi udara dalam ruangan tertutup atau ber-AC. Fan ini dapat dipasang dalam sistem pendeteksi polusi udara berbasis IoT yang menggunakan sensor untuk mendeteksi kualitas udara dan mengatur fan untuk mengurangi polusi. Misalnya, dalam penelitian yang menggunakan sensor MQ-2 dan MQ-135, fan dapat dikontrol melalui mikrokontroler Arduino Uno dan modul WiFi untuk mengirimkan data ke aplikasi, yang dapat memberikan informasi yang akurat mengenai kualitas udara dalam ruangan dan mengatur fan untuk mengurangi polusi (Fajar B et al., (2023).

Salah satu jenis kipas polusi udara adalah kipas DC yang sering digunakan dalam sistem pemanasan, seperti dinding Trombe komposit yang dibahas dalam studi tersebut, di mana kipas DC dipasang di ventilasi atas untuk pengendalian sirkulasi udara yang stabil. Studi tersebut menemukan bahwa penggunaan kipas DC dalam dinding Trombe komposit menghasilkan penurunan beban panas sebesar 27,3% dibandingkan dengan dinding Trombe klasik dan penurunan sebesar 32,1% dibandingkan dengan kasus tanpa dinding Trombe. Namun, efisiensi dalam memanaskan suhu ruangan tidak tinggi tanpa pemanasan (Zhu et al., (2022).

Prinsip kerja Kipas DC tergantung pada sistem yang digunakan. Misalnya, dalam sistem pengendalian motor DC menggunakan Arduino Uno pada rancang bangun electrostatic precipitator, kipas DC akan bekerja selama 40 detik setelah saklar rangkaian control di onkan, dan kecepatannya dapat diatur dengan cara memutar potensio meter yang ada pada ruang controller. Setelah 40 detik, kipas DC akan mati dan motor hammer akan bekerja selama 20 detik untuk merontokan debu yang menempel pada plat CE (Wicaksono & Aribowo, (2019).

### 2.2.5 Internet of Things

Internet of Things (IoT) adalah istilah yang muncul dengan pengertian sebuah akses perangkat elektronik melalui media internet atau tanpa internet. Hal ini merujuk pada penggunaan teknologi yang memungkinkan objek seperti perangkat elektronik, peralatan, atau perangkat lunak untuk mengirim dan menerima data secara otomatis tanpa perlu interaksi manual. IoT memungkinkan perangkat elektronik untuk berinteraksi satu sama lain dan dengan sistem informasi, memungkinkan pengguna untuk mengumpulkan dan menganalisis data dari berbagai sumber (Bella, (2023).

Cara kerja Internet of Things (IoT) dengan mengirimkan dan menerima data dari perangkat keras yang terhubung ke jaringan internet. Setiap perangkat di beri alamat Internet protocol (IP) dan di koneksikan dengan alamat IP perangkat lain sehingga dapat bertukar data. IoT membawa dampak yang sangat baik untuk institusi pemerintah, industri, swasta, pendidikan, kesehatan dan bahkan kalangan rumah (Wilianto & Kurniawan, (2018).

Penggunaan IoT (Internet of Things) dalam kehidupan sehari-hari memiliki beberapa manfaat. IoT dapat digunakan untuk mengoptimalkan penggunaan energi di rumah, sehingga dapat mengurangi penggunaan energi. IoT juga dapat membantu dalam pengambilan keputusan sehari-hari melalui prediksi dan analisis data (Pulungan & Yahfizham, (2023).

### 2.2.6 ThingSpeak

Thingspeak adalah platform online yang mengizinkan pengguna untuk mengumpulkan, mengambil, dan menganalisis data dari perangkat lunak IoT (Internet of Things) dan sensor. Platform ini menyediakan fitur-fitur seperti pengumpulan data, visualisasi data, dan pengaturan alert. Thingspeak dapat digunakan untuk pengkajian data dari perangkat lunak IoT dan sensor, seperti sensor suhu, sensor kelembapan, dan sensor lainnya (Harmoko, 2023).

Penggunaan ThingSpeak sebagai Platform Internet of Things (IoT) memiliki kelebihan dan kekurangan. ThingSpeak memungkinkan pengumpulan dan pemantauan data secara real-time dari berbagai sensor (Chandrasekaran et al., (2023). Platform ini memiliki kekurangan yaitu membutuhkan server untuk mengumpulkan dan mengirim data, yang dapat membuat pengaturan lebih rumit (Salam & Alexander, (2023).

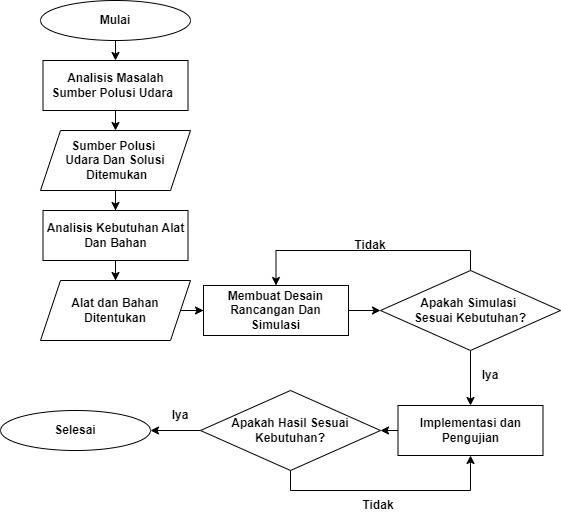
Penggunaan Thingspeak pada monitoring polusi udara dapat dilakukan melalui pengembangan sistem monitoring polusi udara yang terintegrasi dengan platform IoT. Contohnya, dalam penelitian yang dilakukan pada 2023, dirancang sebuah alat terintegrasi dengan platform IoT untuk memantau kualitas dan tingkat polusi udara sebagai solusi untuk buruknya kualitas udara. Alat ini menggunakan dua sensor utama, yaitu sensor debu PM2.5 GP2Y1010AU0F dan sensor suhu dan kelembapan DHT22. Semua sensor dihubungkan ke platform IoT melalui mikrokontroler nodemcu dengan modul Wifi esp8266. Data dari sensor ditampilkan secara real-time pada platform dan dapat dipantau melalui smartphone atau perangkat desktop dengan koneksi internet yang memadai. Platform ini juga dapat mengirimkan notifikasi tentang kondisi udara di sekitar sesuai dengan nilai ambang batas PM2.5 BMKG (Aghorru & Koprawi, 2023).

# BAB III

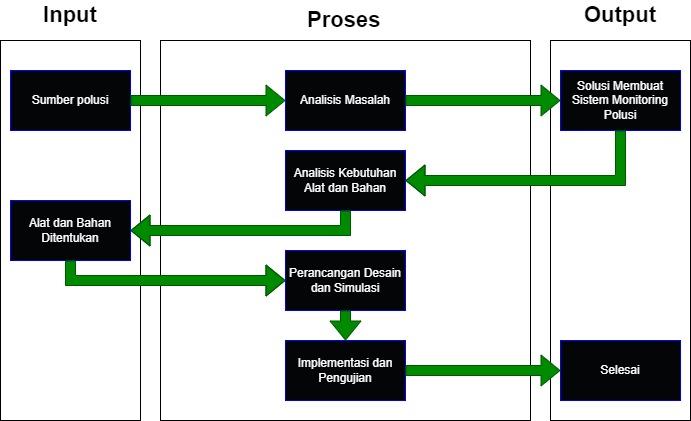
**METODOLOGI PENELITIAN**

## 3.1 Langkah Penelitian

Penelitian ini akan melibatkan serangkaian langkah yang sistematis untuk merancang dan mengimplementasikan sistem monitoring polusi udara berbasis Internet of Things (IoT). berikut adalah Diagram dari langkah penelitian



Gambar 3. 1 Flowchart Langkah Penelitian



Gambar 3. 2 Diagram Blok Langkah Penelitian

Gambar 3.1 adalah Flowchart langkah penelitian dalam melakukan penelitian sistem monitoring polusi udara berbasis Internet of Things (IoT) sedangkan, gambar 3.2 adalah langkah penelitian dalam bentuk diagram blok berikut deskripsi:

### 3.1.1 Analisis Masalah Sumber Polusi Udara

Analisis masalah sumber polusi udara di terminal bus merupakan tahap kritis dalam penelitian sistem monitoring polusi udara berbasis Internet of Things (IoT). Dalam langkah ini, peneliti memeriksa berbagai sumber pencemaran udara yang dapat ditemui di terminal bus. Faktor-faktor seperti kendaraan bus menjadi fokus utama untuk diidentifikasi dampaknya terhadap kualitas udara. Analisis mendalam terhadap jenis polutan yang dihasilkan, dilakukan untuk memahami kebutuhan alat dan bahan yang digunakan untuk melakukan perancangan.

### 3.1.2 Analisis Kebutuhan Alat dan Bahan

Analisis kebutuhan alat dan bahan merupakan langkah penting dalam pengembangan sistem monitoring polusi udara berbasis Internet of Things (IoT). Pada tahap ini, peneliti melakukan evaluasi terhadap peralatan dan komponen yang diperlukan untuk mendukung sistem. Hal ini mencakup pemilihan sensor-sensor yang dapat mengukur berbagai parameter kualitas udara, seperti konsentrasi partikulat, gas berbahaya. Selain itu, penelitian juga mencakup pemilihan perangkat keras dan perangkat lunak yang sesuai untuk mendukung konektivitas IoT, pengolahan data, serta antarmuka pengguna. Teknologi yang digunakan dan kebutuhan aplikatifnya menjadi fokus utama dalam analisis ini.

### 3.1.3 Membuat Desain Rancangan dan Simulasi

Proses pembuatan desain rancangan dan simulasi menjadi tahap selanjutnya dalam pengembangan sistem monitoring polusi udara berbasis Internet of Things (IoT). Pada langkah ini, peneliti merinci keseluruhan sistem, termasuk penempatan sensor dan komponen pendukung. penelitian ini juga mencakup simulasi untuk menguji sejauh mana desain tersebut dapat memberikan respons secara real-time terhadap perubahan kualitas udara. Simulasi tersebut memainkan peran penting dalam pengecekan kinerja sistem sebelum implementasi di lapangan, memungkinkan peneliti untuk mengidentifikasi potensi masalah dan melakukan perbaikan sebelum sistem diimplementasikan secara penuh.

### 3.1.4 Implementasi dan Pengujian

Langkah selanjutnya dalam penelitian sistem monitoring polusi udara berbasis Internet of Things (IoT) adalah implementasi dan pengujian sistem yang telah dirancang. Pada tahap implementasi, peneliti menerapkan secara praktis desain rancangan yang telah disusun sesuai dengan spesifikasi yang telah ditetapkan sebelumnya. Proses ini mencakup instalasi sensor, pemasangan perangkat keras dan perangkat lunak, serta konfigurasi jaringan komunikasi IoT. Selama implementasi, penting untuk melakukan pemantuan antara semua komponen sistem guna memastikan pengumpulan data yang akurat.

Setelah implementasi, langkah selanjutnya adalah melakukan pengujian secara menyeluruh terhadap sistem. Pengujian ini melibatkan evaluasi performa sensor, kestabilan koneksi, serta keberhasilan sistem dalam mengumpulkan, menyimpan, dan mengirimkan data secara real-time. Pengujian juga mencakup simulasi kondisi lingkungan yang berbeda untuk memastikan bahwa sistem dapat mengatasi variasi polusi udara yang mungkin terjadi. Hasil dari pengujian ini memberikan gambaran tentang sejauh mana sistem dapat memenuhi tujuan pengembangan dan apakah perlu dilakukan perbaikan atau peningkatan lebih lanjut.

## 3.2 Alat dan Bahan

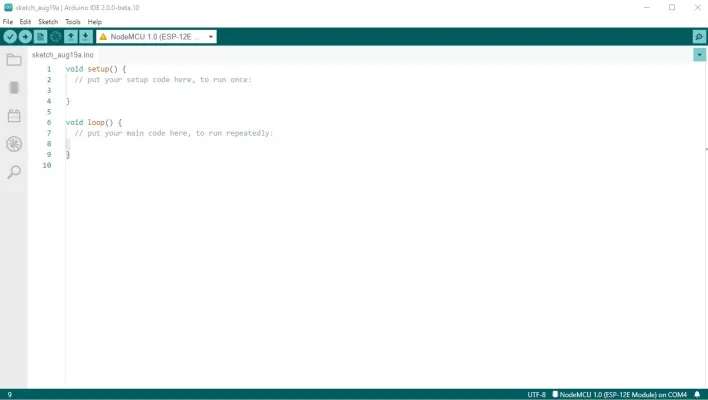
dalam merancang pendeteksi polusi udara berbasis Internet of Things diperlukan Alat dan bahan sebagai berikut:

### 3.2.1 Software

Perangkat lunak (software) adalah elemen inti dunia komputasi yang mengontrol fungsionalitas dan pengoperasian komputer, perangkat seluler, atau perangkat berbasis teknologi lainnya. Perangkat lunak dapat dibagi menjadi dua kategori utama: perangkat lunak sistem dan perangkat lunak aplikasi. Perangkat lunak sistem, seperti sistem operasi, mengelola sumber daya perangkat keras dan menyediakan antarmuka ke aplikasi. Perangkat lunak aplikasi, di sisi lain, dirancang untuk tujuan tertentu, seperti pengolah kata, spreadsheet, desain grafis, atau pengembangan web. Perangkat lunak ini memungkinkan pengguna untuk melakukan berbagai tugas, mulai dari tugas sehari-hari hingga pemrosesan dan analisis data yang kompleks. Kemajuan dalam dunia perangkat lunak terus memungkinkan inovasi, efisiensi, dan kemudahan penggunaan di banyak bidang kehidupan, bisnis, dan penelitian. Di era digital yang terus berkembang, perangkat lunak memainkan peran penting dalam membentuk cara kita berinteraksi dengan teknologi dan dunia di sekitar kita. Adapun beberapa software yang digunakan dalam sistem monitoring polusi udara berbasis Internet of Things (IoT) sebagai berikut:

1. Arduino IDE

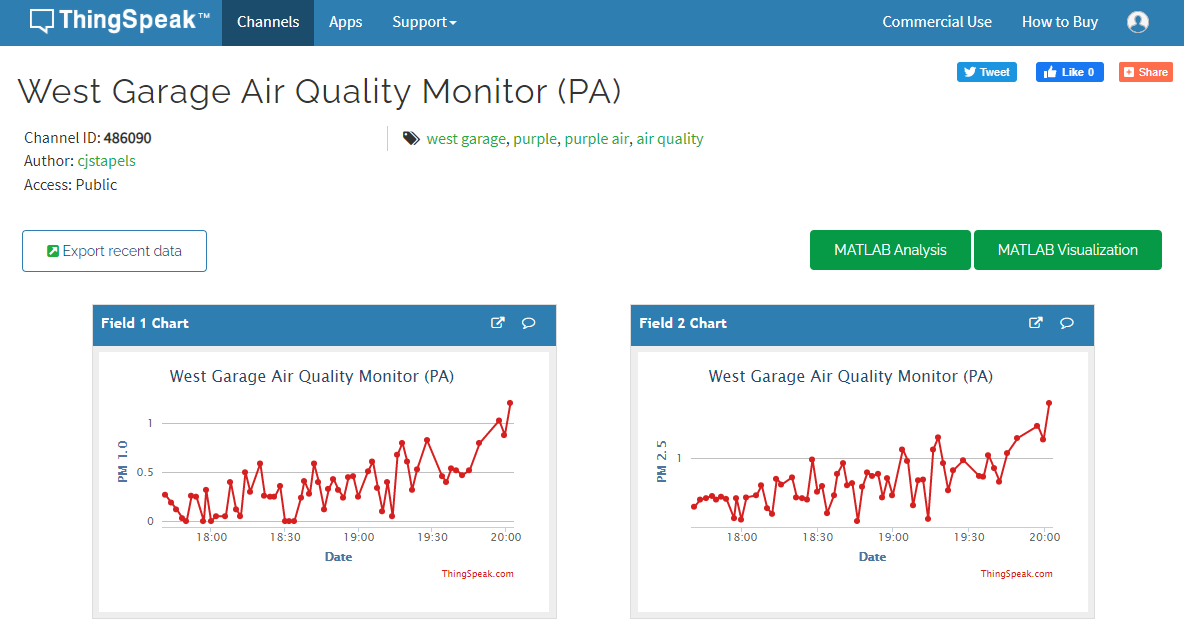
Arduino IDE merupakan perangkat lunak yang kritis dalam pengembangan sistem monitoring polusi udara berbasis Internet of Things (IoT), yang memungkinkan penyesuaian ambang batas dari sensor MQ-135 dan GP2Y1014AU0F untuk mengaktifkan kipas pembersih udara serta koneksi ke platform ThingSpeak. Melalui Arduino IDE, pengguna dapat dengan mudah mengatur parameter ambang batas berdasarkan nilai-nilai yang diinginkan untuk gas dan partikel debu dalam udara, sehingga kipas dapat diaktifkan secara otomatis ketika konsentrasi polutan melebihi batas yang ditentukan. Selain itu, Arduino IDE juga menyediakan alat yang diperlukan untuk menghubungkan sistem ke platform ThingSpeak, di mana data kualitas udara dapat disimpan dan ditampilkan secara real-time.



Gambar 3. 3 Software Arduino IDE

1. ThingSpeak

ThingSpeak adalah platform yang digunakan untuk memonitoring tingkat polusi udara berbasis Internet of Things (IoT). Dengan menggunakan sensor-sensor yang terhubung ke perangkat IoT, seperti sensor gas dan sensor partikulat, ThingSpeak memungkinkan pengguna untuk mengumpulkan data tentang kualitas udara secara real-time. Data-data ini kemudian dapat dianalisis, divisualisasikan, dan dibagikan secara online melalui grafik-grafik dan diagram-diagram yang tersedia dalam platform tersebut. Pengguna juga dapat mengatur pemberitahuan atau alarm jika tingkat polusi udara melebihi batas tertentu. Dengan demikian, ThingSpeak memberikan solusi yang efektif untuk memantau dan mengelola polusi udara secara efisien, serta memberikan wawasan yang berharga bagi masyarakat dan pemangku kepentingan dalam upaya menjaga kualitas udara yang lebih baik.



Gambar 3. 4 Tampilan Platform ThingSpeak

### 3.2.2 Hardware

Perangkat keras komputer mencakup komponen fisik yang membentuk setiap aktivitas di komputer. Ini terdiri dari prosesor (CPU), yang bertanggung jawab untuk menangani perhitungan, dan memori (RAM), yang digunakan untuk menyimpan data sementara saat komputer beroperasi. Selain itu, harddisk atau SSD (Solid State Drive) juga digunakan untuk menyimpan data secara permanen. Komputer juga memiliki kartu grafis yang mengontrol tampilan serta perangkat input seperti keyboard, mouse, dan monitor yang memungkinkan pengguna berinteraksi dengan sistem. Selain itu, terdapat perangkat jaringan seperti kartu jaringan untuk menghubungkan ke Internet, serta berbagai port seperti USB, HDMI dan lain-lain yang mendukung koneksi ke perangkat eksternal. Perkembangan teknologi telah mendorong perangkat keras komputer menjadi lebih cepat, lebih andal, dan lebih efisien dibandingkan sebelumnya, sehingga memungkinkannya melakukan aplikasi dan tugas yang semakin kompleks, mulai dari pekerjaan manufaktur hingga hiburan dan penelitian ilmiah. Perangkat keras berkualitas yang memenuhi kebutuhan Anda sangat penting untuk mendukung kinerja komputasi yang optimal. Adapun hardware yang digunakan dalam sistem monitoring polusi udara berbasis Internet of Things (IoT) sebagai berikut:

1. Laptop

Tabel 3. 1 Spesifikasi Laptop

|  |  |
| --- | --- |
| Komponen | Spesifikasi |
| Tipe | Lenovo 81H6 |
| Processor | Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz |
| RAM: | DDR4-2133 MHz 4,00 GB |
| GPU | Intel(R) HD Graphic 620/NVIDIA Geforce MX130 |
| Sistem operasi | Windows 10 Pro 64 bit |
| Penyimpanan | RX7 2.5 128 GB SSD |

1. Smartphone

Tabel 3. 2 Spesifikasi SmartPhone

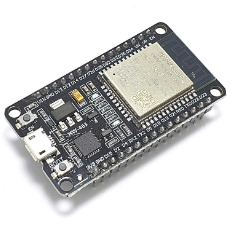
|  |  |
| --- | --- |
| Komponen | Spesifikasi |
| Tipe | Samsung Galaxy A20 |
| Tipe layar | Super AMOLED capacitive touchscreen, 16 juta warna |
| Ukuran | 6.4 inci |
| Resolusi | 720 x 1560 piksel |
| Sistem Operasi | Android 9.0 (Pie) dengan antarmuka pengguna One UI |
| Chipset | Exynos 7884 (14 nm) |
| CPU | Octa-core (2x1.6 GHz Cortex-A73 & 6x1.35 GHz Cortex-A53) |
| GPU | Mali-G71 MP2 |
| RAM | 3 GB |
| Slot MicroSD | Ya, hingga 512 GB (bersama dengan slot SIM kedua) |

### 3.2.3 Komponen Elektronika dan Mikrokontroler

Komponen elektronik dan mikrokontroler merupakan jantung dari banyak sistem elektronik modern. Komponen elektronik, seperti resistor, kapasitor, transistor, dan sirkuit terpadu, menjadi fondasi sirkuit elektronik. Mereka digunakan untuk mengontrol arus listrik, menyimpan energi, dan memperkuat sinyal. Dalam konteks ini, mikrokontroler merupakan otak dari banyak perangkat elektronik. Mikrokontroler adalah semikonduktor yang memiliki unit pemrosesan pusat, memori, dan beberapa pin I/O yang digunakan untuk mengontrol berbagai perangkat. Mereka digunakan dalam berbagai aplikasi, mulai dari perangkat rumah pintar hingga kendaraan otonom, dan membantu memantau, mengontrol, dan mengotomatisasi berbagai tugas. Kombinasi komponen elektronik dan mikrokontroler memungkinkan peneliti merancang perangkat yang cerdas dan efisien untuk memenuhi berbagai kebutuhan. Dengan kemajuan teknologi, kami terus melihat peningkatan kinerja dan kemampuan komponen-komponen tersebut, sehingga membuka jalan bagi inovasi yang lebih maju di masa depan. Adapun beberapa komponen yang digunakan dalam perancangan pendeteksi polusi udara berbasis Internet of Things (IoT) sebagai berikut:

1). ESP32

ESP32 adalah mikrokontroler yang memiliki kemampuan terhubung ke jaringan internet melalui WiFi. Fungsi dari mikrokontroler ini dalam sistem monitoring polusi udara berbasis Internet of Things adalah sebagai pusat kontrol dari komponen sensor dan kipas polusi udara yang dikonfigurasi untuk mengurangi tingkat polusi udara. Dengan menggunakan ESP32 data dari sensor dapat di transmisikan ke server melalui koneksi WiFi yang memungkinkan pengguna dapat memantau tingkat polusi udara secara real-time dari mana saja. Selain itu, ESP32 dapat dikonfigurasi untuk respon dan tindakan cepat untuk mengurangi tingkat polusi udara.



Gambar 3. 5 ESP32

Berikut adalah spesifikasi ESP32 yang digunakan untuk sistem monitoring polusi udara berbasis Internet of Things

Tabel 3. 3 Spesifikasi ESP32

|  |  |
| --- | --- |
| Komponen | Spesifikasi |
| Mikrokontroler | Dual-core Tensilica LX6 dengan kecepatan hingga 240 MHz |
| Memori | Flash Memory internal 520 KB atau 4 MB |
| WiFi | 802.11 b/g/n (2.4 GHz) dan 802.11 a/n/ac (5 GHz), hingga 150 Mbps. |
| Tegangan | 3.3V |

2) Sensor MQ-135

Sensor MQ-135 adalah yang dapat mendeteksi berbagai macam gas beracun yang dapat memmpengaruhi kualitas udara. Dalam sistem monitoring polusi udara berbasis Internet of Things sensor MQ-135 berperan penting dalam mendeteksi polusi udara yang datanya dapar dikirim ke server melalui jaringan internet serta dapat dipantau oleh pengguna melalui aplikasi. Namun ada beberapa hal yang perlu diperhatikan dari sensor MQ-135 yaitu cenderung untuk mendeteksi gas-gas tertentu seperti karbon dioksida (CO2), karbon monoksida (CO), amonia (NH3), metana (CH4), dan senyawa organik volatil (VOCs) yang mungkin tidak seakurat sensor khusus untuk setiap jenis polusi udara. Akan tetapi sensor ini dapat memberikan informasi mengenai tingkat polusi udara dan keberadaan gas-gas yang berpotensi merugikan kesehatan, sehingga pengguna dapat mengambil tindakan cepat untuk mengurangi tingkat polusi udara.



Gambar 3. 6 Sensor MQ-135

Berikut adalah spesifikasi sensor MQ-135 yang digunakan untuk sistem monitoring polusi udara berbasis Internet of Things.

Tabel 3. 4 Spesifikasi Sensor MQ-135

|  |  |
| --- | --- |
| Komponen | Spesifikasi |
| Rentang deteksi | 1. Karbon Monoksida (CO): 10 hingga 1000 ppm 2. Amonia (NH3): 10 hingga 300 ppm 3. Metana (CH4): 1000 hingga 10000 ppm 4. Karbon Dioksida (CO2): 10 hingga 5000 ppm 5. Senyawa Organik Volatil (VOCs) |
| Tegangan | 5V DC |
| Suhu | -10°C hingga 50°C |
| Kelembapan | 20% hingga 90% RH |

3) Sensor GP2Y1014AU0F

Sensor Particulate matter dengan diameter kurang dari 2,5 mikrometer dapat mengukur tingkat polusi udara perkotaan yang berasal dari asap kendaraan, industri dan aktivitas pembakaran lainnya. Fungsi dari sensor GP2Y1014AU0F adalah mengidentifikasi partiker partikel seperti debu halus dan partikel partikel kecil yang dapat membahayakan kesehatan yang memungkinkan pemantauan secara real-time. Prinsip kerja dari sensor GP2Y1014AU0F yaitu mendeteksi partikel debu lalu memberikan informasi ke aplikasi untuk pemantauan.



Gambar 3. 7 Sensor GP2Y1014AU0F

Berikut adalah spesifikasi sensor GP2Y1014AU0F yang digunakan untuk sistem monitoring polusi udara berbasis Internet of Things

Tabel 3. 5 Spesifikasi Sensor GP2Y1014AU0F

|  |  |
| --- | --- |
| Komponen | Spesifikasi |
| Rentang deteksi | 0 hingga 25 µg/m³ |
| Tegangan | 5V DC |
| Suhu | -10°C hingga 50°C |
| Kelembapan | 20% hingga 90% RH |

4) Fan DC

Fan DC memiliki peran penting dalam sistem monitoring polusi udara berbasis Internet of Things yaitu sebagai sistem ventilasi untuk meningkatkan sirkulasi udara di sekitar perangkat monitoring. Prinsip kerja dari fan DC yaitu ketika sensor MQ-135 dan GP2Y1014AU0F mendeteksi adanya polusi yang melebihi ambang batas maka pengguna dapat memberikan tindakan cepat dengan cara mengaktifkan kipas polusi udara. Fan DC dapat diaktifkan melalui aplikasi sesuai kebutuhan sehingga dapat menjaga kesehatan lingkungan terutama dalam tingkat kualitas udara. Dengan demikian, fan DC juga berkontribusi dalam menjaga kinerja sistem monitoring polusi udara berbasis Internet of Things



Gambar 3. 8 Fan DC

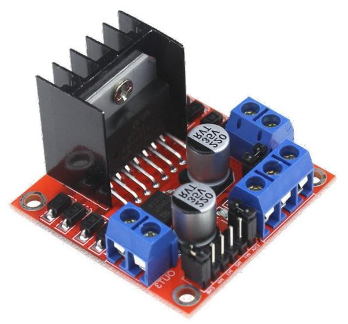
Berikut adalah spesifikasi fan DC yang digunakan untuk sistem monitoring polusi udara berbasis Internet of Things

Tabel 3. 6 Spesifikasi Fan DC

|  |  |
| --- | --- |
| Komponen | Spesifikasi |
| Tegangan | 12V DC |
| Arus listrik | 0.2A (200mA) |
| Daya | 2.4W |
| Kecepatan putaran (RPM) | 0 RPM hingga 1500 RPM |
| Aliran udara | 30 CFM |

5) Driver Motor

Dalam sistem monitoring polusi udara berbasis IoT driver motor berperan penting dalam mengaktifkan kipas sesuai dengan tingkat polusi udara yang dideteksi oleh sensor MQ-135 dan GP2Y1014AU0F. Ketika sensor mendeteksi tingkat polusi yang nilainya melebihi batas yang ditentukan maka driver motok akan menyalurkan daya untuk mengakifkan fan DC untuk menetralisir udara. Selain itu, driver motor juga dapat mengatur kecepatan rotasi kipas polusi udara sesuai dengan tingkat yang dibutukan.



Gambar 3. 9 Driver Motor L298N

Berikut adalah spesifikasi Driver motor yang digunakan untuk sistem monitoring polusi udara berbasis Internet of Things

Tabel 3. 7 Spesifikasi Driver Motor

|  |  |
| --- | --- |
| Komponen | Spesifikasi |
| Tegangan maximum | 12V - 24V DC. |
| Arus maximum | 4A |

7) Kabel jumper

Kabel jumper beefungsi sebagai penghubung dua atau lebih komponen elektronika dalam satu rangkaian. Pada Sistem monitoring polusi udara berbasis IoT kabel jumper berfungsi sebgai penghubung ESP32 dengan sensor MQ-135, GP2Y1014AU0, fan DC, LED. Dengan menggunakan kabel jumper Mikrokontroler yang sudah di konfigurasi dapat menjadi pusat control dari komponen elektronika lainnya.



Gambar 3. 10 Kabel Jumper

## 3.3 Rencana Pengujian

Rencana pengujian dijabarkan untuk menguji kinerja sistem monitoring polusi udara yang dikembangkan. Pengujian akan dilakukan dalam beberapa tahap untuk memastikan kelayakan sistem. Berikut adalah langkah-langkah rinci dalam rencana pengujian:

### 3.3.1 Pengujian Sensor

Pada tahap ini, sensor GP2Y1014AU0F dan MQ-135 yang digunakan untuk mengukur Tingkat polusi akan diuji. Pengujian sensitivitas akan melibatkan penempatan sensor di lingkungan simulasi dengan tingkat polusi yang dikontrol. Pengujian akurasi akan dilakukan dengan membandingkan hasil pengukuran sensor. Pengujian dilakukan dengan memaparkan sensor gas, mengevaluasi respons sensor, dan mencatat hasil pengukuran pada setiap tingkatan konsentrasi. Selain itu, pengujian akurasi melibatkan perbandingan hasil pengukuran pada sensor.

### 3.3.2 Pengujian Kipas Polusi Udara

Tujuan utama pengujian ini adalah untuk mengukur tingkat perputaran udara yang dihasilkan oleh kipas, serta kemampuannya dalam menghilangkan partikulat dan senyawa gas yang umumnya ditemukan dalam udara tercemar. Pengujian sensitivitas kipas akan melibatkan kemampuan adaptasinya terhadap kondisi lingkungan yang berbeda. Pengujian performa kipas akan dilakukan dengan mengatur kondisi uji yang mencakup tingkat polusi udara yang telah ditentukan. Analisis data hasil pengujian akan memberikan gambaran tentang sejauh mana kipas mampu mengurangi tingkat polusi udara dan memperbaiki kualitas udara.

### 3.3.3 Pengujian Antarmuka Pengguna

Rencana pengujian antarmuka pengguna untuk sistem yang menggunakan ThingSpeak sebagai platform pemantauan. Fokus pada pengujian ini adalah keakuratan data yang ditampilkan oleh ThingSpeak dan ThingView yaitu kadar CO dan partikel debu. Pengujian antarmuka ini berfungsi untuk memastikan bahwa orang yang bekerja di sekitar terminal dapat memantau dan memahami kualitas udara di terminal.

# BAB IV

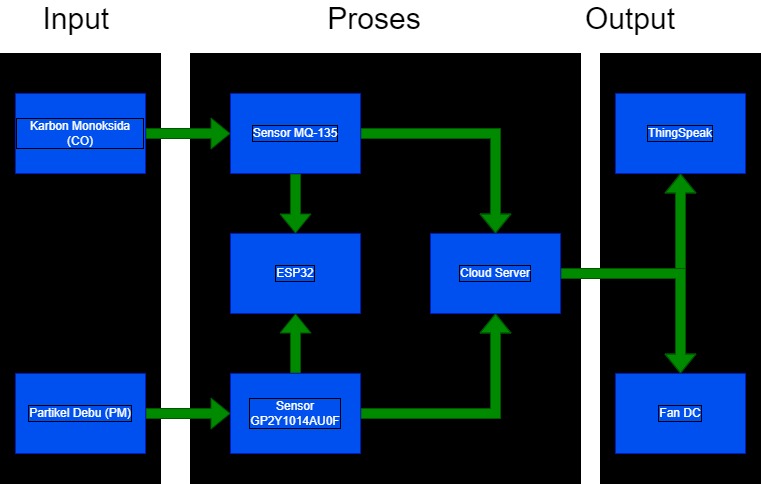
**ANALISIS DAN PERANCANGAN SISTEM**

## 4.1 Rencana Pengembangan Sistem

Dalam penelitian ini membutuhkan Rencana pengembangan sistem yang akan digunakan untuk menyelesaikan penelitian ini. Teknologi Internet of Things ini akan bekerja untuk mendeteksi polusi udara menggunakan sensor MQ-135 dan GP2Y1014AU0F lalu dipantau melalui ThingSpeak dan ThingView. Selain itu, fan DC sebagai kipas untuk menetralisir udara.

### 4.1.1 Blok Diagram Sistem

Pada sistem monitoring polusi udara berbasis Internet of Things terdapat proses blok diagram sebagai berikut:

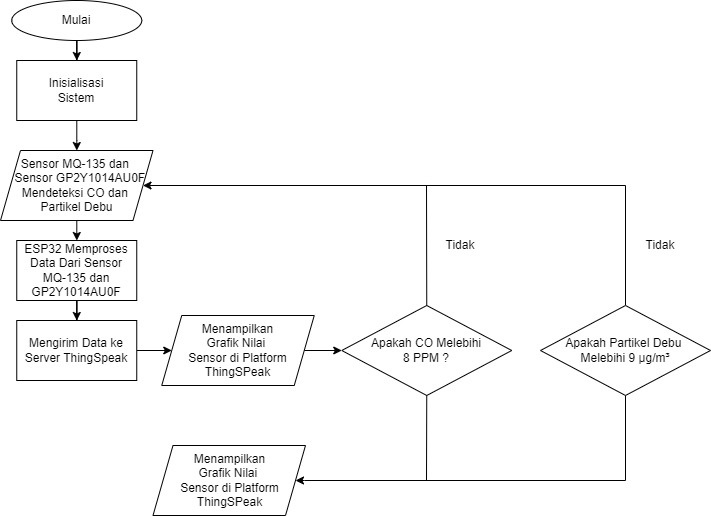


Gambar 4. 1 Blok Diagram Sistem

Gambar 4.1 menggambarkan diagram alur proses pengolahan data sistem monitoring polusi udara berbasis Internet of Things. Pada bagian input, Karbon Monoksida dan Partikel Debu dideteksi oleh Sensor MQ-135 dan Sensor GP2Y1014AU0F secara berturut-turut. Kedua sensor ini terhubung ke ESP32, sebuah modul yang memfasilitasi koneksi internet untuk mengirim data ke server. Di bagian output, data yang telah diolah ditampilkan melalui platform ThingSpeak dan ThingView, serta mengaktifkan Fan DC untuk menetralisir udara.

### 4.1.2 Flowchart Sistem

Pada sistem monitoring polusi udara berbasis Internet of Things terdapat flowchart sebagai berikut:



Gambar 4. 2 Flowchart Sistem

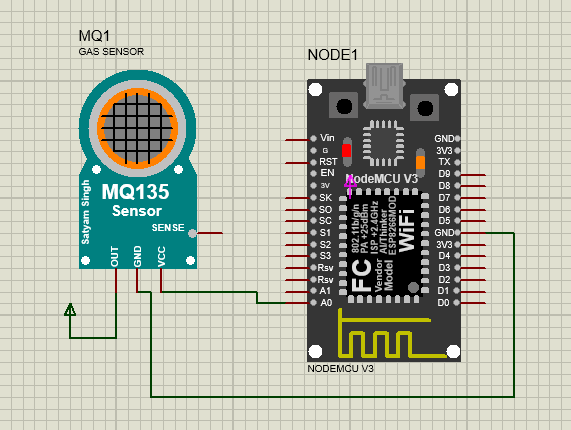
Gambar 4.2 adalah flowchart dari sistem monitoring polusi udara berbasis Internet of Things. Dimulai dari insialisasi sensor lalu ketika sensor MQ-135 dan GP2Y1014AU0F mendeteksi adanya polusi maka data dari polusi di proses oleh ESP32. Setelah data di proses, data akan dikirim ke server cloud lalu ditampilkan pada layout ThingSpeak dan ThingView. Pengecekan juga dilakukan apakah kadar CO melebihi 8 PPM dan partikel debu melebihi 9 µg/m³. Jika melebihi maka sistem akan menjalankan perintah untuk mengaktifkan kipas untuk menetralisir udara.

## 4.2 Perancangan Sistem

Perancangan Sistem adalah tahapan dalam penelitian sistem monitoring polusi udara berbasis Internet of Things sebelum implementasi. Tahapan ini terdiri dari beberapa rangkaian komponen elektronika yang di simulasikan. Berikut adalah perancangan sistem dalam penelitian ini.

### 4.2.1 Rangkaian Sensor MQ-135

Rangkaian ini terdiri dari ESP8266 sebagai mikrokontroler dan MQ-135 sebagai sensor polusi udara.

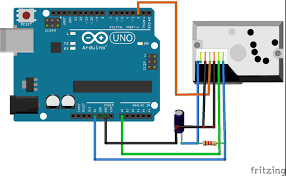


Gambar 4. 3 Rangkaian Sensor MQ-135

Gambar 4.3 secara jelas mengilustrasikan cara menghubungkan sensor gas MQ-135 dengan modul ESP8266 untuk memonitor kualitas udara. Sensor gas MQ-135 ini memiliki dapat mendetksi zat polusi CO. Dengan menggunakan modul ESP8266, sensor dapat diintegerasikan untuk pemantauan secara real-time dan mengatur ambang batas berbahaya untuk mengaktifkan kipas polusi udara. ketika sensor MQ-135 mendeteksi tingkat CO melebihi 8 PPM maka kipas polusi udara akan aktif untuk menetralisir udara

### 4.2.2 Rangkaian Sensor GP2Y1014AU0F

Rangkaian ini terdiri dari Arduino uno sebagai mikrokontroler dan sensor GP2Y1014AU0F sebagai sensor partikel debu.

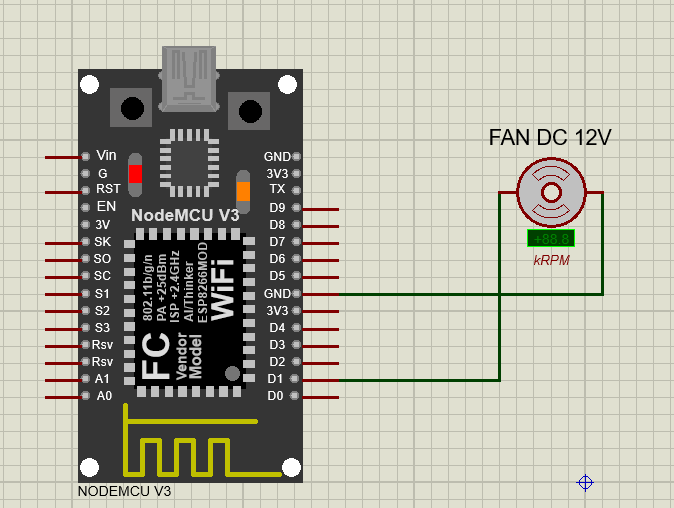


Gambar 4. 4 Rangkaian Sensor GP2Y1014AU0F

Gambar 4.5 adalah rangkaian GP2Y1014AU0F yang disambungkan dengan mikrokontroler arduino uno. Mikrokontroler bertugas sebagai otak sistem yang dapat diprogram untuk mengatur Sensor GP2Y1014AU0F. Sensor GP2Y1014AU0F mendeteksi partikel-partikel kecil dalam udara yang dapat berdampak negatif pada kesehatan manusia jika terhirup dalam jumlah besar. Dengan memanfaatkan sensor GP2Y1014AU0F yang terhubung ke mikrokontroler, dapat diintegrasikan untuk pemantauan secara real-time dan dapat mengatur ambang batas untuk mengaktifkan kipas polusi udara. Ketika tingkat partikel debu melebihi 9 µg/m³ maka kipas polusi udara akan aktif untuk menetralisir udara.

### 4.2.3 Rangkaian Fan DC

Rangkaian ini terdiri dari ESP8266 sebagai mikrokontroler dan fan DC sebagai kipas polusi udara.

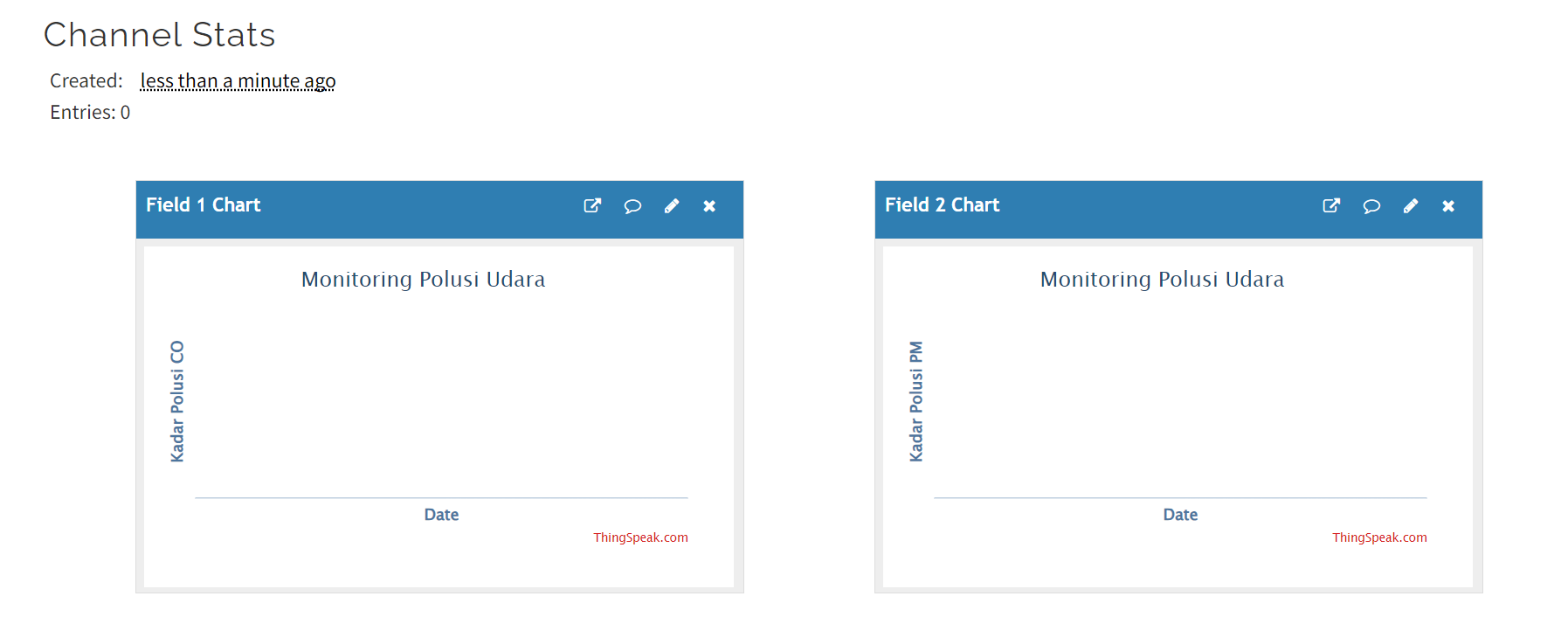


Gambar 4. 5 Rangkaian Fan DC

Gambar 4.4 adalah kipas polusi udara dengan menggunakan papan mikrokontroler ESP8266 dan Fan DC. Pada dasarnya, mikrokontroler ESP8266 berperan sebagai otak sistem, yang dapat diprogram untuk mengontrol fan DC sesuai dengan tingkat polusi udara yang diukur oleh sensor yang sesuai. Ketika MQ-135 dan GP2Y1014AU0F mendeteksi kadar CO melebihi 8 PPM dan partikel debu melebihi 9 µg/m³, mikrokontroler akan memberikan instruksi kepada fan DC untuk meningkatkan sirkulasi udara.

### 4.2.3 Rancangan ThingSpeak

ThingSpeak pada penelitian ini digunakan sebagai aplikasi pemantauan kadar polusi yang dideteksi oleh sensor MQ-135 dan GP2Y1014AU0F yaitu karbon monoksida (CO) dan partikel debu (PM).



Gambar 4. 6 Rancangan Layout ThingSpeak

Gambar 4.5 menampilkan dua grafik kosong yang menawarkan pandangan yang terperinci tentang kualitas udara. Grafik pertama, dengan label "Field 1 Chart," ditujukan untuk memantau polusi dari sensor MQ-135 yaitu kadar karbon monoksida (CO), sedangkan grafik kedua, yang berlabel "Field 2 Chart," bertujuan untuk memantau partikulat materi (PM) yang dideteksi oleh sensor GP2Y1014AU0F. mereka mewakili kemampuan sistem untuk menyajikan informasi penting tentang tingkat polusi udara secara real-time.

# BAB V

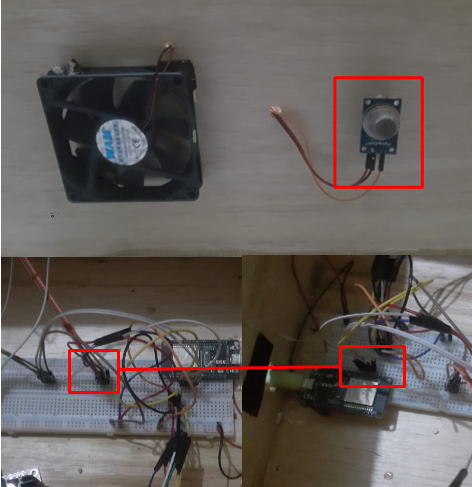
**HASIL DAN PEMBAHASAN**

## 5.1 Impelementasi

Pada penelitian sistem monitoring polusi udara berbasis Internet of Things akan implementasikan sebagai berikut:

### 5.1.1 Implementasi sensor MQ-135

Berikut adalah implementasi sensor MQ-135 yang dihubungkan ke ESP32 pada penelitian sistem monitoring polusi udara berbasis Internet of Things.

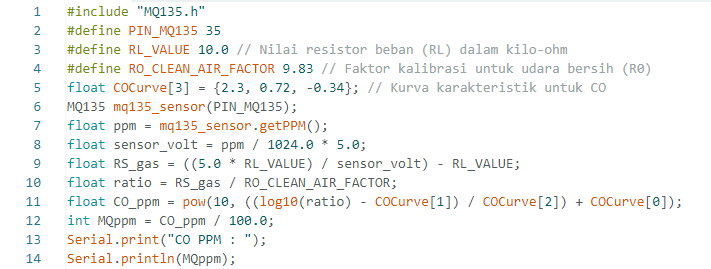


Gambar 5. 1 Implementasi Sensor MQ-135

Gambar 5.1 adalah hasil implementasi sensor MQ-135 yang akan dijelaskan sebagai berikut.

* + - 1. Pin VCC dihubungkan pada pin 3.3V yang berfungsi mengambil daya dari ESP32 untuk sensor
      2. Pin GND dihubungkan pada pin GND ESP32
      3. Pin A0 dihubungkan pada pin GPIO35 ESP32 yang berfungsi untuk membaca nilai analog sensor

Pada implementasi sensor MQ-135 di ESP32 menghasilkan kode program sebagai berikut.



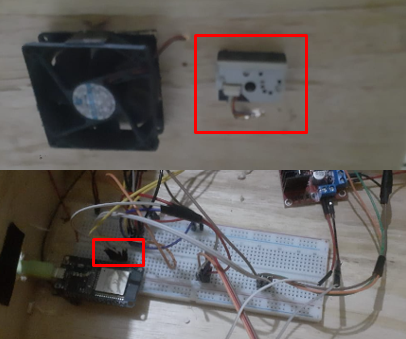
Gambar 5. 2 Kode Program Sensor MQ-135

Gambar 5.2 adalah kode program konfigurasi awal untuk sensor MQ-135 yang akan dijelaskan sebagai berikut.

1. #include "MQ135.h": Memasukkan library MQ135 yang digunakan untuk berinteraksi dengan sensor MQ135.
2. #define PIN\_MQ135 35: Mendefinisikan pin yang terhubung ke sensor MQ135.
3. #define RL\_VALUE 10.0: Mendefinisikan nilai resistor beban (RL) dalam kilo-ohm.
4. #define RO\_CLEAN\_AIR\_FACTOR 9.83: Mendefinisikan faktor kalibrasi untuk udara bersih (R0).
5. float COCurve[3] = {2.3, 0.72, -0.34} : Mendefinisikan kurva karakteristik untuk CO. Ini adalah array dengan tiga elemen yang merepresentasikan koefisien dalam persamaan untuk menghitung ppm CO dari nilai resistansi sensor.
6. MQ135 mq135\_sensor(PIN\_MQ135) : Membuat objek MQ135 dengan menggunakan pin yang telah ditentukan sebelumnya.
7. float ppm = mq135\_sensor.getPPM() : Mendapatkan nilai ppm (parts per million) CO dari sensor MQ135.
8. float sensor\_volt = ppm / 1024.0 \* 5.0 : Menghitung tegangan keluaran sensor.
9. float RS\_gas = ((5.0 \* RL\_VALUE) / sensor\_volt) - RL\_VALUE : Menghitung resistansi sensor dalam kondisi gas.
10. float ratio = RS\_gas / RO\_CLEAN\_AIR\_FACTOR : Menghitung rasio resistansi sensor terhadap resistansi udara bersih.
11. float CO\_ppm = pow(10, ((log10(ratio) - COCurve[1]) / COCurve[2]) + COCurve[0]) : Menggunakan persamaan yang telah didefinisikan sebelumnya untuk menghitung ppm CO.
12. int MQppm = CO\_ppm / 100.0 : Mengonversi nilai ppm CO menjadi bilangan bulat (integer) yang merepresentasikan ratusan ppm.
13. Serial.print("CO PPM : ") : Menampilkan pesan awal untuk output serial.
14. Serial.println(MQppm) : Menampilkan nilai ppm CO pada output serial.

### 5.1.2 Implementasi Sensor GP2Y1014AU0F

Berikut adalah implementasi sensor GP2Y1014AU0F yang dihubungkan ke ESP32 pada penelitian sistem monitoring polusi udara berbasis Internet of Things.

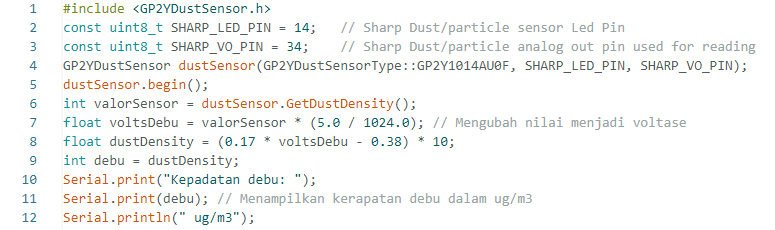


Gambar 5. 3 Implementasi Sensor GP2Y1014AU0F

Gambar 5.3 adalah hasil implementasi sensor GP2Y1014AU0F yang akan dijelaskan sebagai berikut.

* + - 1. Kabel berwarna merah sebagai VCC pada sensor GP2Y1014AU0F dihubungkan pada pin 3.3V untuk mengambil daya dari ESP32 ke sensor.
      2. Kabel berwarna kuning dihubungkan pada pin GPIO34 pada ESP32 untuk membaca nilai analog pada sensor GP2Y1014AU0F
      3. Kabel berwarna hitam, biru, hijau, putih karena tidak digunakan pada sistem ini maka dihubungkan pada pin GND ESP32

Pada implementasi sensor GP2Y1014AU0F di ESP32 menghasilkan kode program sebagai berikut.



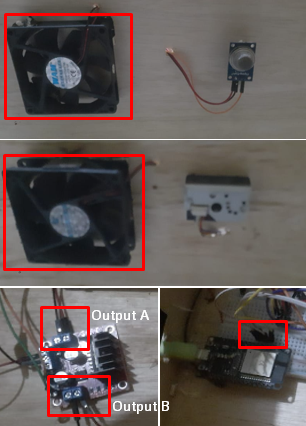
Gambar 5. 4 Kode Program Sensor GP2Y1014AU0F

Gambar 5.4 adalah kode program sensor GP2Y1014AU0F yang akan dijelaskan sebagai berikut.

1. #include <GP2YDustSensor.h>: Memasukkan library GP2YDustSensor yang digunakan untuk mengakses sensor debu Sharp.
2. const uint8\_t SHARP\_LED\_PIN = 14 : Mendefinisikan pin pada mikrokontroler Arduino yang terhubung ke LED pada sensor debu Sharp.
3. const uint8\_t SHARP\_VO\_PIN = 34 : Mendefinisikan pin pada mikrokontroler Arduino yang terhubung ke output analog pada sensor debu Sharp untuk membaca data.
4. GP2YDustSensor dustSensor(GP2YDustSensorType::GP2Y1010AU0F, SHARP\_LED\_PIN, SHARP\_VO\_PIN) : Inisialisasi objek dustSensor dari kelas GP2YDustSensor dengan tipe sensor GP2Y1010AU0F dan pin LED serta pin output analog yang sudah didefinisikan sebelumnya.
5. dustSensor.begin() : Memulai sensor debu untuk melakukan inisialisasi dan persiapan pembacaan.
6. int valorSensor = dustSensor.GetDustDensity() : Membaca nilai kepadatan debu dari sensor dan menyimpannya dalam variabel valorSensor.
7. float voltsDebu = valorSensor \* (5.0 / 1024.0) : Mengonversi nilai pembacaan sensor menjadi tegangan dalam volt dengan membagi nilai pembacaan dengan 1024 (resolusi ADC Arduino) dan mengalikannya dengan 5 volt (tegangan referensi).
8. float dustDensity = (0.17 \* voltsDebu - 0.38) \* 10 : Menghitung kepadatan debu dengan menggunakan rumus yang telah ditentukan oleh datasheet sensor debu Sharp GP2Y1010AU0F, kemudian hasilnya dikalikan dengan 10 untuk mendapatkan nilai dalam ug/m3.
9. int debu = dustDensity : Mengonversi nilai kepadatan debu menjadi tipe data integer dan menyimpannya dalam variabel debu.
10. Serial.print("Kepadatan debu: ") : Mencetak string "Kepadatan debu: " ke serial monitor.
11. Serial.print(debu) : Mencetak nilai kepadatan debu (dalam ug/m3) ke serial monitor.
12. Serial.println(" ug/m3") : Mencetak string " ug/m3" diikuti dengan newline ke serial monitor untuk memberikan format pada output.

### 5.1.3 Implementasi Fan DC

Berikut adalah implementasi fan DC yang dihubungkan ke ESP32 pada penelitian sistem monitoring polusi udara berbasis Internet of Things.

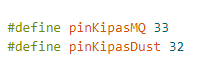


Gambar 5. 5 Implementasi Fan DC

Gambar 5.5 adalah hasil implementasi fan DC yang akan dijelaskan sebagai berikut

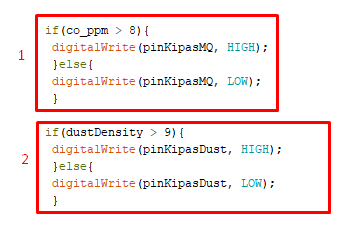
* + - 1. Fan DC 1 dihubungkan pada output A driver motor dan fan DC 2 dihubungkan pada Output B driver motor
      2. Input 1 dihubungkan pada pin GPIO26 ESP32 dan input 2 dihubungkan dengan pin GPIO27 ESP32
      3. Driver motor mengambil tegangan 12V dari adaptor

Pada implementasi Fan DC di ESP32 menghasilkan kode program sebagai berikut.



Gambar 5. 6 Konfigurasi Pin Fan DC

Gambar 5.6 adalah program konfigurasi awal fan DC yaitu mendefinisikan pin fan DC di pin 33 dan 32



Gambar 5. 7 Kode Program Mengaktifkan Fan DC

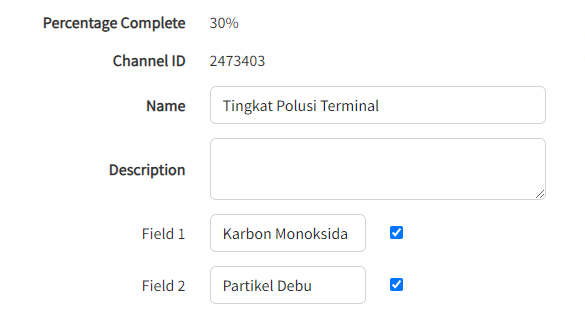
Gambar 5.7 adalah kode program untuk opuput dari fan DC yang akan dijelaskan sebagai berikut.

Jika nilai sensor MQ-135 melebihi 8 PPM maka fan DC 1 aktif jika tidak maka fan DC 1 nonaktif

Jika nilai sensor GP2Y1014AU0F melebihi 9 µg/m³ fan DC 2 aktif jika tidak maka fan DC 2 nonaktif

### 5.1.4 Implementasi ThingSpeak

Berikut adalah implementasi ThingSpeak sebagai media pemantauan dalam sistem monitoring berbasis Internet of Things (IoT)

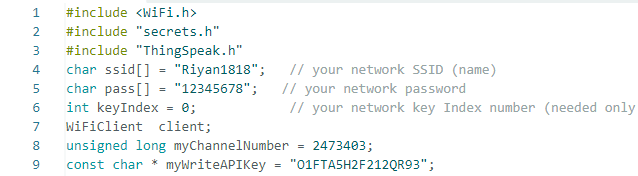


Gambar 5. 8 Pembuatan Channel ThingSpeak

Gambar 5.8 adalah pembuatan channel ThingSpeak yang akan dijelaskan sebagai berikut.

* + - 1. Nama Channel: Tingkat Polusi Terminal
      2. Field 1: Karbon Monoksida, untuk menampilkan nilai CO yang dihasilkan sensor MQ-135
      3. Field 2: Partikel Debu, untuk menampilkan nilai yang dihasilkan oleh sensor GP2Y1014AU0F

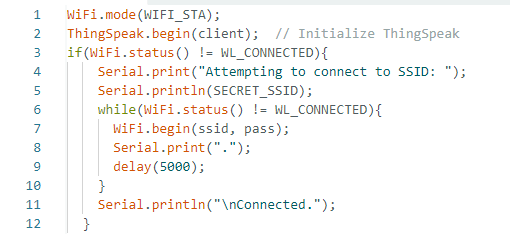
Pada Impelementasi ThingSpeak sebagai media pemantauan menghasilkan kode program sebagai berikut.



Gambar 5. 9 Konfigurasi Awal ThingSPeak

Gambar 5.9 adalah konfigurasi awal Thingspeak pada ESP32 yang akan dijelaskan sebagai berikut

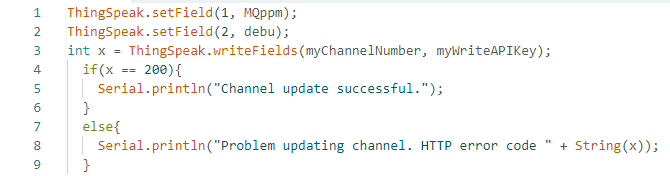
1. #include <WiFi.h>: Memasukkan library WiFi untuk menghubungkan perangkat ke jaringan WiFi.
2. #include "secrets.h": Memasukkan file secrets.h, yang kemungkinan berisi informasi sensitif seperti kredensial jaringan atau API key yang tidak ingin disertakan langsung dalam kode utama untuk alasan keamanan.
3. #include "ThingSpeak.h": Memasukkan library ThingSpeak yang digunakan untuk mengirimkan data ke platform ThingSpeak.
4. char ssid[] = "Riyan1818" // your network SSID (name): Mendefinisikan nama jaringan WiFi (SSID) yang akan digunakan untuk menghubungkan perangkat.
5. char pass[] = "12345678" // your network password: Mendefinisikan kata sandi jaringan WiFi.
6. int keyIndex = 0 // your network key Index number (needed only for WEP): Mendefinisikan indeks kunci jaringan, yang biasanya diperlukan hanya untuk jaringan WEP. Dalam konteks ini, nilai 0 berarti tidak digunakan karena WPA/WPA2 tidak memerlukan indeks kunci.
7. WiFiClient client : Membuat objek WiFiClient yang akan digunakan untuk mengelola koneksi jaringan.
8. unsigned long myChannelNumber = 2473403 : Mendefinisikan nomor saluran ThingSpeak di mana data akan dikirim. Setiap pengguna ThingSpeak memiliki saluran unik untuk mengorganisasi data mereka.
9. const char \* myWriteAPIKey = "O1FTA5H2F212QR93" : Mendefinisikan API key untuk menulis data ke saluran ThingSpeak. API key ini diperlukan untuk mengautentikasi dan mengotorisasi pengiriman data ke saluran tertentu.



Gambar 5. 10 Kode Proram Mengaktifkan ThingSpeak

Gambar 5.10 adalah kode program untuk menghubungkan dan mengaktifkan ThingSpeak pada ESP32

1. WiFi.mode(WIFI\_STA): Mengatur mode WiFi pada perangkat menjadi mode WIFI\_STA (Station mode). Dalam mode ini, perangkat bertindak sebagai klien yang dapat terhubung ke jaringan WiFi yang ada, bukan sebagai titik akses (access point).
2. ThingSpeak.begin(client) // Initialize ThingSpeak: Memulai atau menginisialisasi komunikasi dengan ThingSpeak menggunakan objek client yang telah dibuat sebelumnya. Ini mengatur ThingSpeak untuk siap mengirim atau menerima data melalui koneksi WiFi.
3. if(WiFi.status() != WL\_CONNECTED){: Memeriksa status koneksi WiFi. Jika perangkat belum terhubung ke jaringan WiFi (WiFi.status() tidak sama dengan WL\_CONNECTED), maka blok kode di dalam pernyataan if akan dijalankan.



Gambar 5. 11 Kode Program Field ThingSpeak

Gambar 5.11 adalah kode program untuk menampilkan nilai sensor di field ThingSpeak yang akan dijelaskan sebagai berikut

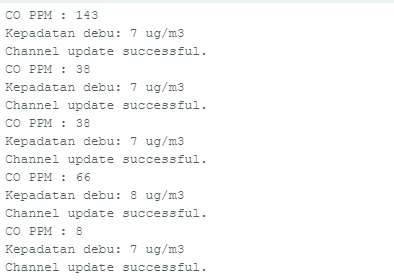
1. ThingSpeak.setField(1, MQppm): Mengatur nilai MQppm ke bidang (field) pertama pada saluran ThingSpeak. MQppm adalah nilai yang akan dikirim ke field 1 di saluran ThingSpeak.
2. ThingSpeak.setField(2, debu): Mengatur nilai debu ke bidang (field) kedua pada saluran ThingSpeak. debu adalah nilai lain yang akan dikirim ke field 2 di saluran ThingSpeak.
3. int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey): Mengirim data yang telah diatur ke saluran ThingSpeak menggunakan nomor saluran (myChannelNumber) dan API key (myWriteAPIKey). Fungsi writeFields mengirimkan semua field yang telah diatur sebelumnya ke ThingSpeak. Fungsi ini mengembalikan kode status HTTP dari server ThingSpeak.

## 5.2 Pengujian Sistem

Berikut adalah hasil pengujian dari sistem monitoring polusi udara berbasis Internet of Things.

### 5.2.1 Pengujian Sensor

Berikut adalah hasil pengujian sensor MQ-135 sensor melalui serial monitor Arduino IDE



Gambar 5. 12 Serial Monitor Pengujian Sensor MQ-135

Gambar 5.12 adalah hasil pengujian sensor MQ-135 mendeteksi karbon monoksida yang akan dijelaskan sebagai berikut.

Tabel 5. 1 Pengujian Sensor MQ-135

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Waktu (Detik) | Nilai Sensor MQ-135  (PPM) | Nilai Referensi MQ-7 (PPM) | Nilai Kesalahan  (PPM) | | Error | Keterangan |
| 1 | 20 | 143 | 172 | 29 | | 16,86% | Berhasil |
| 2 | 40 | 38 | 39 | 1 | | 2,56% | Berhasil |
| 3 | 60 | 38 | 38 | 0 | | 0% | Berhasil |
| 4 | 80 | 66 | 68 | 2 | | 2,94% | Berhasil |
| 5 | 100 | 8 | 8 | 0 | | 0% | Berhasil |
| Rata Rata Nilai Sensor MQ-135 | | | | | 58,6 PPM | | | |
| Rata Rata presentase Kesalahan | | | | | 4,472% | | | |

Tabel 5.1 adalah tabel pengujian sensor MQ-135 mendeteksi karbon monoksida dalam waktu 100 detik. nilai sensor MQ-135 (dalam ppm), nilai referensi MQ-7 (dalam ppm), nilai kesalahan (dalam ppm), dan persentase error untuk beberapa percobaan. Setiap baris mewakili satu percobaan yang dilakukan untuk membandingkan hasil sensor MQ-135 dengan nilai referensi MQ-7. Dari data yang disajikan, terlihat bahwa terdapat lima percobaan yang dilakukan pada interval waktu yang berbeda. Hasilnya menunjukkan bahwa pada percobaan pertama, terdapat kesalahan sebesar 16,86% antara nilai sensor MQ-135 dan nilai referensi MQ-7. Namun, meskipun terdapat kesalahan, percobaan tersebut masih dapat dianggap berhasil. Demikian pula dengan percobaan lainnya, di mana terlihat bahwa kesalahan umumnya rendah, dengan beberapa percobaan mencapai nilai kesalahan 0%, menunjukkan tingkat akurasi yang baik dalam pengukuran sensor MQ-135 terhadap nilai referensi MQ-7.

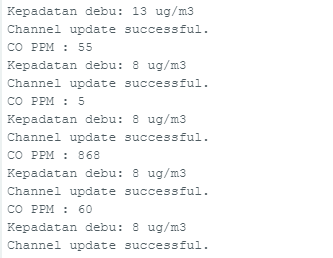
Adapun pengujian jarak deteksi sensor MQ-135 yang akan dijelaskan sebagai berikut

Tabel 5. 2 Pengujian Jarak Sensor MQ-135

|  |  |  |  |
| --- | --- | --- | --- |
| No | Jarak | Konsentrasi CO | Keterangan |
| 1 | 5 cm | 22 PPM | Berhasil |
| 2 | 10 cm | 11 PPM | Berhasil |
| 3 | 15 cm | 5 PPM | Berhasil |
| 5 | 20 cm | 1 PPM | Berhasil |

Tabel 5.2 adalah hasil pengujian jarak deteksi sensor MQ-135 untuk mengukur konsentrasi CO pada berbagai jarak. Pada jarak 5 cm, sensor mendeteksi konsentrasi gas sebesar 22 PPM dan dinyatakan berhasil. Ketika jarak ditingkatkan menjadi 10 cm, nilai yang terdeteksi turun menjadi 11 PPM, dan pengukuran ini juga dinyatakan berhasil. Pada jarak 15 cm, konsentrasi gas yang terukur adalah 5 PPM dengan hasil pengukuran yang tetap berhasil. Terakhir, pada jarak 20 cm, sensor mendeteksi konsentrasi gas sebesar 1 PPM dan pengujian ini juga berhasil. Data tersebut menunjukkan bahwa sensor MQ-135 masih mampu mendeteksi keberadaan gas dengan baik hingga jarak 20 cm, meskipun nilai PPM yang terdeteksi menurun seiring dengan peningkatan jarak.

Adapun pengujian Sensor GP2Y1014AU0F yang akan ditampilkan dalam serial monitor Arduino IDE sebagai berikut.



Gambar 5. 13 Serial Monitor Sensor GP2Y1014AU0F

Gambar 5.13 adalah hasil pegujian sensor GP2Y1014AU0F dalam mendeteksi partikel debu yang akan dijelaskan sebagai berikut.

Tabel 5. 3 Pengujian Sensor GP2Y1014AU0F

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Waktu (Detik) | Nilai Sensor | Output | Ketera ngan |
| 1 | 20 | 13 µg/m³ | Kipas Aktif | Berhasil |
| 2 | 40 | 8 µg/m³ | Kipas Nonaktif | Berhasil |
| 3 | 60 | 8 µg/m³ | Kipas Nonaktif | Berhasil |
| 4 | 80 | 8 µg/m³ | Kipas Nonaktif | Berhasil |

Tabel 5.2 menampilkan hasil pengujian sensor GP2Y1014AU0F dengan berbagai nilai waktu (dalam detik) dan respons sensor yang terkait. Setiap baris merepresentasikan satu pengujian yang dilakukan pada interval waktu yang berbeda. Hasil pengujian menunjukkan bahwa pada waktu 20 detik, nilai output sensor adalah 13, yang mengindikasikan aktivasi kipas. Percobaan tersebut dianggap berhasil. Sementara pada waktu-waktu berikutnya (40, 60, dan 80 detik), nilai output sensor adalah 8, yang menandakan bahwa kipas dalam keadaan nonaktif. Secara keseluruhan, hasil pengujian menunjukkan bahwa sensor GP2Y1014AU0F memberikan respons yang sesuai dengan parameter yang diharapkan, dengan kipas diaktifkan atau dinonaktifkan sesuai dengan tingkat deteksi sensor terhadap kondisi yang diperiksa.

### 5.2.2 Pengujian Fan DC

Adapun hasil pengujian fan DC sebagai kipas polusi udara akan dijelaskan sebagai berikut.

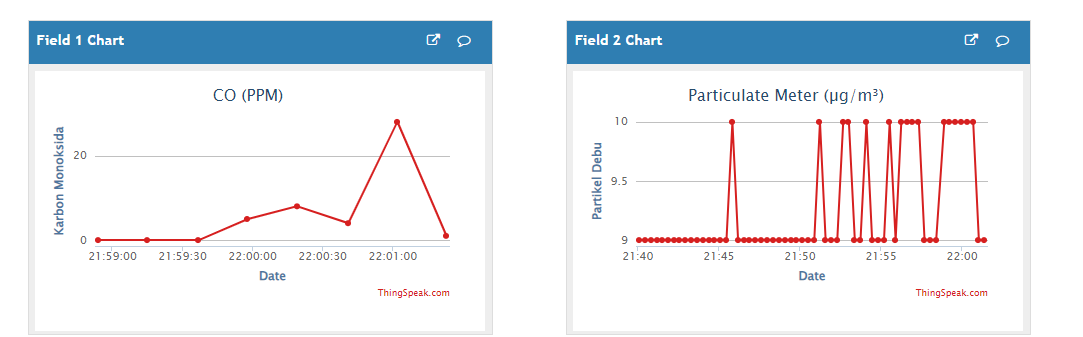
Tabel 5. 4 Pengujian Fan DC

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Sensor | Waktu  (Detik) | Nilai Awal | Nilai Akhir | Output | Keterangan |
| 1 | MQ-135 | 40 | 143 PPM | 38 PPM | Kipas Aktif | Berhasil |
| 2 | MQ-135 | 60 | 143 PPM | 38 PPM | Kipas Aktif | Berhasil |
| 3 | MQ-135 | 80 | 143 PPM | 66 PPM | Kipas Aktif | Berhasil |
| 4 | MQ-135 | 100 | 143 PPM | 8 PPM | Kipas Nonaktif | Berhasil |
| 5 | GP2Y1014AU0F | 40 | 13 µg/m³ | 8 µg/m³ | Kipas Nonaktif | Berhasil |
| 6 | GP2Y1014AU0F | 60 | 13 µg/m³ | 8 µg/m³ | Kipas Nonaktif | Berhasil |
| 7 | GP2Y1014AU0F | 80 | 13 µg/m³ | 8 µg/m³ | Kipas Nonaktif | Berhasil |

Tabel 5.3 adalah hasil penurunan tingkat polusi setelah fan DC aktif. Hasil pengujian menunjukkan bahwa ketika kipas diaktifkan, nilai sensor MQ-135 mengalami penurunan yang signifikan dari nilai awal ke nilai akhir dalam rentang waktu 40, 60, dan 80 detik. Hal ini mengindikasikan bahwa kipas DC berhasil dalam mengurangi tingkat konsentrasi karbon monoksida yang dideteksi oleh sensor MQ-135 dari nilai awal 143 PPM Hingga 8 PPM. Demikian pula, hasil pengujian pada sensor GP2Y1014AU0F menunjukkan pengurangan nilai sensor dari 13 µg/m³ menjadi 8 µg/m³ ketika kipas di aktifkan dalam rentang waktu yang sama. Dengan demikian, pengujian ini mengkonfirmasi efektivitas penggunaan kipas DC dalam mengurangi tingkat nilai sensor pada kedua jenis sensor yang diuji.

### 5.2.3 Pengujian ThingSpeak

Adapun pengujian platform ThingSpeak sebagai media pemantauan tingkat polusi udara berikut hasilnya.



Gambar 5. 14 Grafik Hasil Pengujian ThingSpeak

Gambar 5.14 adalah tampilan pengujian platform ThingSpeak yang akan dijelaskan sebagai berikut

Tabel 5. 5 Pengujian ThingSpeak

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Komponen yang Diuji** | **Metode Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Keterangan** |
| 1 | Grafik Data Sensor MQ135 | Melihat grafik data dari sensor MQ135 di ThingSpeak | Grafik menampilkan data CO dengan benar | Grafik tampil sesuai | Berhasil |
| 2 | Grafik Data Sensor GP2Y1014AU0F | Melihat grafik data dari sensor GP2Y1014AU0F di ThingSpeak | Grafik menampilkan data debu dengan benar | Grafik tampil sesuai | Berhasil |
| 3 | Tampilan Data Real-time | Memantau data sensor di ThingSpeak dalam waktu nyata | Data diperbarui secara berkala sesuai interval pengiriman | Data diperbarui | Berhasil |

Tabel 5.4 adalah hasil pengujian ThingSpeak dalam menampilkan grafi dari kedua sensor. Pengujian mencakup lima aspek utama: koneksi sensor MQ135 dan GP2Y1014AU0F ke ESP32 dan ThingSpeak, tampilan data real-time, serta visualisasi data sensor dalam bentuk grafik di ThingSpeak. Hasil pengujian menunjukkan bahwa data dari kedua sensor berhasil dikirim dan ditampilkan di ThingSpeak. Selain itu, data diperbarui secara berkala sesuai dengan interval pengiriman yang diharapkan. Grafik data dari sensor MQ135 dan GP2Y1014AU0F juga berhasil menampilkan informasi karbon monoksida dan kepadatan debu dengan benar, sehingga keseluruhan sistem bekerja dengan baik sesuai dengan yang diharapkan.

Pengujian keakuratan nilai sensor pada platform ThingSpeak akan dijelaskan sebagai berikut

Tabel 5. 6 Pengujian Keakuratan Grafik ThingSpeak

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Waktu | Sensor | Nilai Sensor | Keterlambatan | Keterangan |
| 1 | 21 : 59 : 37 | MQ-135 | 0 ppm | 0 Detik | Berhasil |
| 2 | 21 : 59 : 58 | MQ-135 | 5 ppm | 1 Detik | Berhasil |
| 3 | 22 : 00 : 19 | MQ-135 | 8 ppm | 1 Detik | Berhasil |
| 4 | 22 : 00 : 41 | MQ-135 | 4 ppm | 2 Detik | Berhasil |
| 5 | 22 : 01 : 02 | MQ-135 | 28 ppm | 1 Detik | Berhasil |
| 6 | 22 : 01 : 23 | MQ-135 | 1 ppm | 1 Detik | Berhasil |
| 7 | 21 : 59 : 37 | GP2Y1014AU0F | 10 µg/m³ | 0 Detik | Berhasil |
| 8 | 21 : 59 : 58 | GP2Y1014AU0F | 10 µg/m³ | 1 Detik | Berhasil |
| 9 | 22 : 00 : 19 | GP2Y1014AU0F | 10 µg/m³ | 1 Detik | Berhasil |
| 10 | 22 : 00 : 41 | GP2Y1014AU0F | 10 µg/m³ | 2 Detik | Berhasil |
| 11 | 22 : 01 : 02 | GP2Y1014AU0F | 9 µg/m³ | 1 Detik | Berhasil |
| 12 | 22 : 01 : 23 | GP2Y1014AU0F | 9 µg/m³ | 1 Detik | Berhasil |

Tabel 5.5 menampilkan hasil pengujian keakuratan grafik menggunakan platform ThingSpeak. Setiap pengukuran pada tabel ini menunjukkan bahwa data berhasil diunggah ke ThingSpeak dengan keterlambatan yang bervariasi antara 0 hingga 2 detik, yang masih dalam batas toleransi yang dapat diterima. Misalnya, pada pukul 21:59:37, sensor MQ-135 mencatat nilai 0 ppm tanpa keterlambatan, dan sensor GP2Y1014AU0F mencatat nilai 10 µg/m³ juga tanpa keterlambatan. Pada waktu berikutnya, terjadi keterlambatan 1 hingga 2 detik, tetapi semua data berhasil diunggah dengan akurat.

# BAB VI

**PENUTUP**

## 6.1 Kesimpulan

Adapun beberapa kesimpulan yang dapat diambil dalam penelitian ini yang akan dijelaskan sebagai berikut.

* + - 1. Sistem ini dilengkapi dua sensor MQ-135 dan GP2Y1014AU0F untuk mendeteksi karbon monoksida dan partikel debu serta platform ThingSpeak sebagai media pemantauan yang menampilkan grafik hasil deteksi sensor secara real-time sehingga dapat melalakukan deteksi dini tingkat polusi udara.
      2. Sistem ini dilengkapi dengan dua fan DC pada dua titik sensor sebagi kipas polusi udara yang dapat mengurangi tingkat karbon monoksida dan partikel debu dalam waktu antara 1 sampai 2 menit.
      3. Dengan sistem ini orang yang bekerja di sekitar terminal dapat memantau dan mengatatasi masalah polusi udara yang dihasilkan oleh asap kendaraan di terminal bus

## 6.2 Saran

Adapun Saran untuk pengembangan sistem dalam penelitian ini yang akan dijelaskan sebagai berikut.

Sebaiknya sistem ini dikembangkan lebih lanjut dengan menambah jenis sensor untuk mendeteksi berbagai macam polutan lainnya selain karbon monoksida dan partikel debu

Menyempurnakan antarmuka pengguna pada platform ThingSpeak agar lebih informatif. Penggunaan visualisasi data yang lebih menarik dan mudah dipahami akan membantu pengguna dalam memantau kualitas udara.

# DAFTAR PUSTAKA

A’yun, I. Q., & Umaroh, R. (2023). Polusi Udara dalam Ruangan dan Kondisi Kesehatan: Analisis Rumah Tangga Indonesia. *Jurnal Ekonomi Dan Pembangunan Indonesia*, *23*(1), 16–26.

Aghorru, R., & Koprawi, M. (2023). Rancang Bangun Sistem Pemantau Kualitas Dan Polusi Udara Pm2.5 Yang Terintegrasi Dengan Platform Iot. *Technologia: Jurnal Ilmiah*, *14*(3), 311.

Bella, S. (2023). Implementasi Smart Akuarium Berbasis Internet of Things (IoT) Pada Salma Akuarium Ikan Hias. *Jurnal Teknik Informatika Kaputama (JTIK)*, *7*(2), 322–330.

Bhalli, J. A., Khan, Q. M., Haq, N., Khaliq, A., & Nasir, M. (2016). Cytogenetic assessment of road side plant bioassay for air pollution monitoring: A case study. Environmental Science and Pollution Research, 23(12), 11706-11717.

Budihartono, E., & Afriliana, I. (2019). Monitoring Ketinggian Plateau Berbasis Mikrokontroler Menggunakan Atmega 328 Dan Sensor Altimeter. *Sebatik*, *23*(2), 440–446.

Chandrasekaran, G., Kumar, N. S., Chokkalingam, A., Gowrishankar, V., Priyadarshi, N., & Khan, B. (2023). IoT enabled smart solar water heater system using real time ThingSpeak IoT platform. *IET Renewable Power Generation*, *May*, 1–13.

Crnosija, N., Zamora, M. L., Rule, A. M., & Payne-Sturges, D. (2022). Laboratory Chamber Evaluation of Flow Air Quality Sensor PM2.5 and PM10 Measurements. *International Journal of Environmental Research and Public Health*, *19*(12).

Fachrizal, F., Julham, J., & Antoni, A. (2022). Sistem Monitoring Polusi Udara Mengggunakan Sensor Nitrogen Carbon Berbasis Internet of Thing. *Jurnal Sistem Komputer Dan Informatika (JSON)*, *4*(1), 199.

Fajar B, M., Lestary, F. D., Hidayat, A., Fadhilatunisa, D., & Eka, A. (2023). Prototype Sistem Monitoring Pendeteksi dan Penyaringan Udara pada Ruangan Berbasis Internet of Things (IoT). *Jurnal MediaTIK*, *6*(2), 1.

Fikri, A., Darmawan, I. A., & Fatkhurrohman, M. (2023). Rancang Bangun Monitoring Kadar Polusi Udara di Lingkungan Kampus FKIP Menggunakan Sistem IoT. *JIIP - Jurnal Ilmiah Ilmu Pendidikan*, *6*(5), 2931–2935.

Gito Resmi, M., Hafid, M., & Zulpria Awaludin, S. (2023). Metode Technique for Order of Preference By Similarity To Ideal Solution Dan Mikrokontroler Node Mcu Pada Pendukung Keputusan Pemilihan Jenis Tanah Budidaya Tanaman Aquascape. *Jurnal Mnemonic*, *6*(2), 102–107.

Hakim, T. N., & Susanto, M. F. (2020). Sistem Monitoring Kualitas Udara Berbasis Internet of Things. *Prosiding The 11th Industrial Research Workshop and National Seminar*, *1*, 26–27.

Hamid, A., Rahmadya, B., & Anandika, A. (2023). Rancang Bangun Perangkap Tikus Jenis Falling Floor Trap Menggunakan Dengan Notifikasi Android Berbasis Mikrokontroler. *Chipset*, *4*(02), 119–128.

Harmoko, D. D. (2023). Apa itu Copywriting? (Memahami Definisi Copywriting dari Sudut Pandang Lain). *Literatus*, *5*(2), 253–259.

Humairoh, G. P., & Putra, R. D. E. (2021). Prototipe Pengendalian Kualitas Udara Indoor Menggunakan Mikrokontroler dengan Sensor MQ135, DHT-22 dan Filter HEPA. *Jurnal Serambi Engineering*, *7*(1), 2529–2536.

Manalu, J. W., & Gunoto, P. (2023). Perancangan Sistem Monitoring Kecepatan Angin Dan Temprature Udara Berbasis Internet of Things (Iot). *Sigma Teknika*, *6*(1), 086–096.

Mauluddin, Y., Cahyadi, U., & Maria, M. (2023). Perencanaan Pengendalian Polusi Udara pada Lingkungan Kerja KN. Collection. *Jurnal Kalibrasi*, *21*(2), 92–98.

Nur Kurnia Fibiani, & Rini puji astutik. (2023). Sistem Monitoring Kosentrasi Amonia (NH3) Pada Kawasan PT. Petrokimia Gresik Berbasis IoT. *Journal Zetroem*, *5*(2), 203–205.

Pendriadi, P., Meliala, S., Muthalib, M. A., & BIntoro, A. (2023). Studi Kadar Gas Amonia Menggunakan Sensor Amonia Mq135 Menggunakan Spreadsheet Berbasis Internet of Thing (Iot). *Transmisi: Jurnal Ilmiah Teknik Elektro*, *25*(2), 75–84.

Pulungan, M. K. A., & Yahfizham. (2023). Menjelajahi Dunia Algoritma Pemrograman: Membuka Potensi Baru Dalam Kehidupan Sehari-Hari. *Jurnal Pengabdian Masyarakat Dan Riset Pendidikan*, *2*(2), 33–39.

Ramli, H. R., & Arief, L. (2021). Sistem Otomatisasi Plant Factory dengan Tiga Jenis Tanaman Sayuran Berbeda Berbasis Mikrokontroler dan Android. *Chipset*, *2*(01), 20–32.

Salam, F., & Alexander, O. (2023). Perancangan Monitoring Suhu Berbasis Internet of Things Dengan Node Mcu Esp8266, Dht 11 Dan Thingspeak. *Jurnal Ilmiah Informatika*, *11*(01), 22–26.

Salamah, I., Tapera, R., & Hadi, I. (2022). Alat Penjernih Udara dengan Sensor Radar RCWL dan Monitoring PM2.5 Berbasis IoT. *JTEV (Jurnal Teknik Elektro Dan Vokasional)*, *8*(2), 349.

Suharti, N., & Br. Surbakti, K. (2020). Pemanfaatan Tanaman Blanceng (Dieffenbachia Spp) Sebagai Penurunan Angka Mikroorganisme Di Udara Pada Ruang Tunggu Terminal Bus Als Amplas Medan. *Jurnal Ilmiah PANNMED (Pharmacist, Analyst, Nurse, Nutrition, Midwivery, Environment, Dentist)*, *15*(2), 182–186.

Susilo, T., & David, F. (2023). Sistem Pemantauan Gas Berbahaya Pada Peternakan Ayam Berbasis Internet of Things. *IT-Explore: Jurnal Penerapan Teknologi Informasi Dan Komunikasi*, *2*(3), 247–257.

Wicaksono, M. H. D., & Aribowo, W. (2019). Pengendalian Motor DC Menggunakan Arduino Uno Pada Rancang Bangun Electrostatic Precipitator. *INAJEEE Indonesian Journal of Electrical and Eletronics Engineering*, *2*(2), 63–67.

Wilianto, & Kurniawan, A. (2018). Sejarah, Cara Kerja Dan Manfaat Internet of Things. *Matrix*, *8*(2), 36–41.

Yassin, A., Yebra-Pimentel, E., García-Bereguir, M. B., & Sanchis, E. (2005). Minimizing Worker Exposure to Crystalline Silica in Stone Countertop Manufacturing. Journal of Occupational and Environmental Hygiene, 2(7), 359-368.

Zhu, Y., Zhang, T., Ma, Q., & Fukuda, H. (2022). Thermal Performance and Optimizing of Composite Trombe Wall with Temperature-Controlled DC Fan in Winter. *Sustainability (Switzerland)*, *14*(5). https://doi.org/10.3390/su14053080

Zulem, N. R., Tony, M., & Utomo, S. (2015). Perbandingan Konsumsi Bahan Bakar Antara Bus Dan Travel Minibus Rute Semarang-Solo Sampai Tahun 2040 Menggunakan Software Leap. *Jurnal Teknik Mesin*, *3*(2), 157–162.