

Desain Sistem Peringatan Kualitas Udara Menggunakan NodeMCU dan *Platform IoT*

Rangga Bimo Hermawan¹, Setiyono^{2,*}

¹Fakultas Ilmu Komputer dan Teknologi Informasi/Sistem Komputer, Universitas Gunadarma

²Fakultas Teknologi Industri/Teknik Elektro, Universitas Gunadarma

^{1,2}Jln. Margonda Raya 100 Pondok Cina Depok, Indonesia

¹ranggabimo261102@gmail.com

*Korespondensi: setiyono@staff.gunadarma.ac.id

Abstract— Increased air pollution in various regions necessitates an air quality monitoring system that provides real-time information easily accessible to the public. This study discusses the design of an air quality warning system based on NodeMCU and the Internet of Things (IoT) platform as a simple, inexpensive, and efficient monitoring solution. This system is designed with gas sensors (MQ-series) to detect pollutant concentrations, a NodeMCU ESP8266 module for data processing and network connection, and the Blynk platform IoT to display data online in the form of graphs, notifications, and air quality status indicators. Test results show that the system is capable of displaying real-time changes in air quality with a sufficient level of accuracy compared to reference measuring instruments. The system can also send automatic warning notifications when air quality exceeds certain thresholds. The limitations of this study include sensor sensitivity to the environment and data delays when the internet connection is unstable. This study is still limited to the use of MQ-135, DHT22, and LDR sensors, while international air quality indicators include PM2.5, PM10, CO, NO₂, SO₂, and O₃. Therefore, additional dust particle and specific gas sensors are needed to make the system more representative of WHO standards, leaving room for further development to make it more informative and comprehensive.

Keywords— air quality, NodeMCU ESP8266, Internet of Things (IoT), warning system, real-time monitoring

Abstrak— Peningkatan polusi udara di berbagai wilayah menuntut adanya sistem pemantauan kualitas udara yang mampu memberikan informasi secara real-time dan mudah diakses oleh masyarakat. Penelitian ini membahas desain sistem peringatan kualitas udara berbasis NodeMCU dan *platform* Internet of Things (IoT) sebagai solusi pemantauan yang sederhana, murah, dan efisien. Sistem ini dirancang dengan sensor gas (MQ-series) untuk mendeteksi konsentrasi polutan, modul NodeMCU ESP8266 sebagai pengolah data dan penghubung ke jaringan, serta *platform IoT* Blynk untuk menampilkan data secara daring dalam bentuk grafik, notifikasi, dan indikator status kualitas udara. Hasil pengujian menunjukkan bahwa sistem mampu menampilkan perubahan kualitas udara secara *real-time* dengan tingkat akurasi yang memadai dibandingkan alat ukur referensi. Sistem juga dapat mengirimkan notifikasi peringatan otomatis saat kualitas udara melewati ambang batas tertentu. Keterbatasan penelitian ini mencakup sensitivitas sensor terhadap lingkungan dan keterlambatan data saat koneksi internet tidak stabil. Penelitian ini masih terbatas pada penggunaan sensor MQ-135, DHT22, dan LDR, sementara indikator kualitas udara internasional mencakup PM2.5, PM10, CO, NO₂, SO₂, dan O₃. Oleh karena itu, diperlukan penambahan sensor partikel debu dan gas spesifik agar sistem lebih representatif terhadap standar WHO sehingga masih terbuka peluang pengembangan agar lebih informatif dan lengkap.

Kata Kunci— kualitas udara, NodeMCU ESP8266, Internet of Things (IoT), sistem peringatan, pemantauan real-time

I. PENDAHULUAN

Kualitas udara merupakan salah satu indikator penting dalam menjaga kesehatan lingkungan dan manusia. Peningkatan jumlah kendaraan bermotor, aktivitas industri, dan pembakaran terbuka secara signifikan telah menyebabkan peningkatan konsentrasi polutan udara menjadi asam di berbagai wilayah perkotaan seperti Jakarta, Surabaya, Medan masuk ke dalam wilayah dengan kondisi udara buruk atau tidak sehat [1], [2]. Polusi udara, khususnya yang mengandung partikel halus, seperti PM2.5, dan gas berbahaya, seperti CO, NO₂, dan SO₂, telah terbukti berkontribusi terhadap meningkatnya kasus penyakit pernapasan, kardiovaskular, serta menurunkan kualitas hidup masyarakat [3],[4]. Dalam konteks Indonesia, pencemaran udara telah menjadi isu nasional yang memerlukan penanganan, pemantauan, dan penegakan hukum dengan metode *multidoor system* bagi pelaku pencemaran lingkungan serta penanganan lain yang berkelanjutan [5], [6].

Sistem pemantauan pencemaran udara yang tersedia saat ini masih terbatas pada lokasi-lokasi tertentu yang diduga mengalami penurunan kualitas udara [7]. Oleh karena itu, diperlukan solusi alternatif yang bersifat *real-time*, terjangkau, dan mudah diakses oleh masyarakat [8]. Perkembangan teknologi *Internet of Things* (IoT) menawarkan solusi inovatif dalam pemantauan kualitas udara [9]. Dengan mengintegrasikan sensor lingkungan, mikrokontroler, dan konektivitas internet, sistem IoT dapat mengumpulkan, mengirimkan, dan menampilkan data kualitas udara secara langsung dan dapat berinteraksi kepada pengguna [10] [11]. Salah satu perangkat yang populer dalam implementasi sistem IoT adalah NodeMCU yang mendukung komunikasi Wi-Fi dan kompatibel dengan berbagai jenis sensor gas, seperti MQ135, DHT11, dan lainnya [12].

Berbagai penelitian telah mengembangkan sistem *monitoring* berbasis IoT untuk berbagai aplikasi lingkungan. Misalnya, studi oleh Roudhotul [13] menunjukkan bahwa sistem berbasis NodeMCU dan sensor MQ135 mampu mendeteksi konsentrasi gas CO dan memberikan notifikasi melalui *platform IoT*. Sementara itu, Kemahyanto [14] berhasil mengintegrasikan sensor dengan ThingSpeak untuk pemantauan *real-time* kualitas udara dalam ruang tertutup. Penelitian lain oleh Edi Susilo [15] merancang sistem deteksi gas berbahaya menggunakan sensor terintegrasi yang dapat digunakan di area industri. Pengembangan sistem peringatan kualitas udara juga dikaji oleh Yunus Tjandi [16] yang merancang sistem peringatan dini berbasis IoT untuk mendeteksi peningkatan konsentrasi gas berbahaya. Ade Sumaedi [17] menyoroti pentingnya pemilihan sensor dan desain mikrokontroler dalam meningkatkan akurasi sistem. Pandu [18] mengembangkan prototipe pemantauan lingkungan berbasis NodeMCU dan sensor DHT22 untuk pemantauan kelembapan dan suhu sebagai parameter pendukung kualitas udara. Sementara itu, Alberth Pariama [19] menerapkan pendekatan sistem terdistribusi menggunakan jaringan IoT untuk pelaporan kualitas udara menggunakan fitur notifikasi pada aplikasi Telegram. Yuli Wahyuni [20] merancang alat monitoring kualitas udara yang dapat diakses melalui perangkat seluler dan terhubung dengan layanan *cloud* untuk pengolahan data.

Secara garis besar penelitian sebelumnya pada dasarnya dilakukan pada kondisi area dengan/parameter pemilihan sensor tertentu, sedangkan penelitian ini mengembangkan kajian pada kondisi/parameter terhadap waktu, suhu, dan area yang mengandung gas berbahaya sehingga dapat memberikan perspektif baru. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem peringatan kualitas udara berbasis NodeMCU dan *platform IoT* yang mampu mendeteksi parameter kualitas udara, mengirimkan data secara *real-time*, serta memberikan notifikasi peringatan apabila parameter melebihi ambang batas yang telah ditentukan. Motivasi utama penelitian ini adalah meningkatnya permasalahan polusi udara yang berdampak langsung pada kesehatan masyarakat sehingga diperlukan sistem pemantauan kualitas udara secara *real time* yang mampu memberikan informasi akurat dan terkini.

Penelitian ini berkontribusi dalam pengembangan sistem peringatan kualitas udara berbasis NodeMCU dan IoT yang mampu menampilkan data secara *real-time* sekaligus memberikan notifikasi

peringatan otomatis kepada pengguna. Kontribusi utama terletak pada desain sistem yang sederhana, hemat biaya, namun tetap andal sehingga dapat dijadikan sebagai solusi alternatif untuk pemantauan lingkungan pada skala rumah tangga maupun komunitas. Selain itu, penelitian ini juga membuka peluang integrasi dengan *platform* data lingkungan yang lebih luas untuk mendukung program *smart city* dan peningkatan kesadaran masyarakat terhadap kesehatan lingkungan.

II. METODOLOGI

A. Tahapan Penelitian

Metode penelitian ini diawali dengan perancangan sistem peringatan kualitas udara berbasis NodeMCU dan IoT. Sistem dirancang untuk membaca data kualitas udara menggunakan sensor gas (seri MQ), kemudian mengirimkan data secara nirkabel ke *platform* IoT Blynk melalui modul Wi-Fi pada NodeMCU ESP8266. Data ditampilkan dalam bentuk grafik, indikator kualitas udara, dan notifikasi peringatan.

Pada bagian perangkat keras (*hardware*), komponen utama yang digunakan meliputi NodeMCU ESP8266 sebagai mikrokontroler dan penghubung internet; sensor MQ-135 untuk mendeteksi gas berbahaya; sensor DHT11 untuk mengukur suhu dan kelembaban sebagai faktor lingkungan yang memengaruhi pembacaan sensor gas; dan catu daya 5V untuk operasional sistem. Pada bagian perangkat lunak (*software*) digunakan Blynk *platform* IoT untuk menampilkan data secara *real-time*, menyimpan *log data*, dan mengirimkan notifikasi peringatan ke pengguna melalui *smartphone*.

Prosedur penelitian ini adalah sebagai berikut:

1. Perancangan skematik dan rangkaian sistem yang menggunakan NodeMCU, sensor, dan modul pendukung.
2. Pemrograman NodeMCU untuk membaca data sensor dan mengirimkannya ke server Blynk.
3. Pengujian fungsionalitas sistem dengan memberikan variasi kondisi udara (lingkungan normal, asap kendaraan, asap rokok) untuk melihat respon sensor.
4. Validasi hasil pembacaan sensor dengan membandingkan hasil sistem dengan alat referensi standar (*air quality meter*).
5. Analisis performa sistem dari segi akurasi pembacaan, stabilitas pengiriman data, dan kecepatan respon notifikasi.

B. Perancangan dan Implementasi Sistem

Perancangan sistem dilakukan secara modular dengan setiap komponen sensor terhubung ke mikrokontroler. Data yang dikumpulkan dari sensor diproses dan ditampilkan di *platform* ThingSpeak secara *real-time*. Rangkaian juga dilengkapi dengan *buzzer* dan LED sebagai indikator peringatan lokal. Konfigurasi perangkat keras DHT22 terhubung ke pin D5 untuk membaca suhu serta kelembaban ruangan. MQ-135 terhubung ke pin A0 untuk mendeteksi konsentrasi gas, seperti CO₂. LDR dipasang pada pin D0 untuk mendeteksi kondisi terang atau gelap. LED pada pin D1 dan *buzzer* pada pin D2 untuk memberikan notifikasi visual dan suara.

ESP8266 dikonfigurasi untuk terhubung ke jaringan WiFi dan mengirimkan data ke ThingSpeak setiap 15 detik. ThingSpeak digunakan untuk memvisualisasikan data dan menyimpan informasi kondisi lingkungan secara historis. Penggunaan algoritme *decision tree* dan *k-nearest neighbor* (KNN) dimanfaatkan untuk mengklasifikasikan kondisi udara berdasarkan data dari sensor. Model AI ini dibuat di Python dan diuji untuk menghasilkan label klasifikasi: Sehat, Kurang Sehat, atau Bahaya. Model kemudian

digunakan untuk menentukan kondisi lingkungan secara otomatis. Gambar 1 memperlihatkan diagram blok sistem peringatan kualitas udara menggunakan NodeMCU dan platform IoT.

Gambar 1 merupakan diagram sistem yang dibangun secara keseluruhan. Blok aktivator (*power supply*) merupakan sumber daya utama yang digunakan untuk memberikan tegangan ke seluruh komponen dalam sistem. Blok *input* sistem terdiri dari sensor-sensor yang bertanggung jawab untuk mengumpulkan data lingkungan secara *real-time*. Sensor-sensor tersebut adalah sensor LDR untuk mengukur tingkat pencahayaan dalam ruangan, sensor MQ-135 untuk mendeteksi kualitas udara atau keberadaan gas berbahaya, dan sensor DHT22 untuk mengukur suhu dan kelembaban udara.

Blok Arduino Uno yang digunakan pada sistem ini adalah untuk memproses data dari semua sensor; menjalankan logika kecerdasan buatan (AI) untuk klasifikasi kondisi ruangan; mengirimkan data ke *cloud platform IoT (ThingSpeak/Blynk)*; dan mengaktifkan output jika diperlukan (misalnya: alarm atau LED). Blok *cloud* yang menggunakan platform IoT, yaitu ThingSpeak, adalah untuk menerima data dari ESP8266 melalui koneksi WiFi, menyimpan dan menampilkan data dalam bentuk grafik secara *real-time, monitoring* jarak jauh oleh pengguna melalui HP atau komputer, dan memberikan notifikasi otomatis atau integrasi dengan analisis AI lanjutan (perangkat lunak pemrosesan sinyal). Blok *output* sistem terdiri dari perangkat yang dikontrol oleh ESP8266 untuk memberikan respons terhadap lingkungan. Blok ini berisi lampu LED menyala jika lingkungan gelap dan lampu LED mati jika lingkungan terang, *buzzer* untuk kondisi lingkungan bahaya, dan hasil analisis pada aplikasi bahwa ruangan tidak sehat.

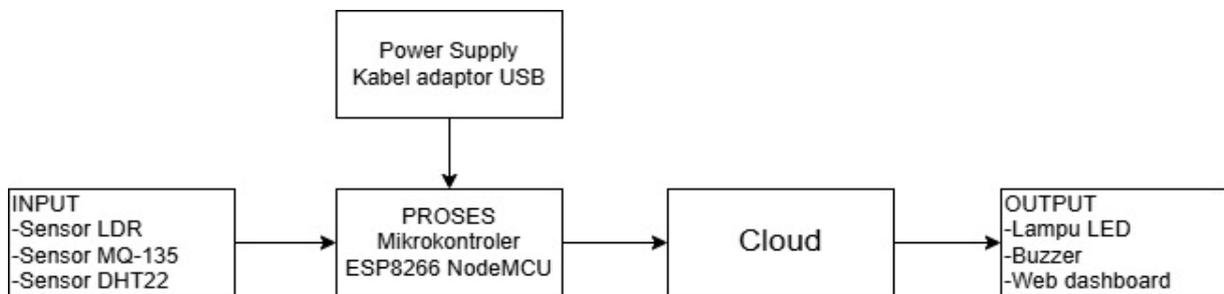
1) Rancangan Yang Diusulkan

Pada tahap ini dirancang sebuah sistem *monitoring* lingkungan cerdas yang mampu membaca data suhu, kelembaban, intensitas cahaya, dan kualitas udara dalam ruangan secara otomatis. Sistem ini terintegrasi dengan teknologi *Internet of Things (IoT)* untuk memungkinkan pengiriman data secara *real-time* ke *cloud*, serta dilengkapi dengan logika kecerdasan buatan (AI) untuk memberikan klasifikasi kondisi lingkungan.

Rancangan sistem ini bertujuan untuk menghasilkan alat monitoring lingkungan yang praktis dan murah; mendeteksi kondisi udara dalam ruangan secara akurat; memberikan respon atau peringatan secara otomatis kepada pengguna jika ditemukan kondisi yang tidak sehat; menyediakan tampilan data *real-time* secara *online* melalui platform IoT.

2) Komponen Yang Digunakan

Rancangan sistem terdiri dari perangkat keras dan perangkat lunak. Perangkat keras terdiri atas ESP8266 NodeMCU (mikrokontroler WiFi), sensor DHT22 (suhu dan kelembaban), sensor MQ-135 (kualitas udara), sensor LDR (intensitas cahaya), LED dan *buzzer* (notifikasi lokal), kabel *jumper* dan *breadboard*, *power supply* (USB). Perangkat lunak terdiri atas Arduino IDE (untuk pemrograman mikrokontroler), platform IoT (ThingSpeak), dan Google Colab atau Matlab (opsional untuk pelatihan AI).



Gambar 1 Diagram blok sistem

3) Metode Pengumpulan dan Pemrosesan Data Yang Digunakan

Sensor membaca kondisi lingkungan setiap beberapa detik. Nilai sensor dikirim ke ESP8266 kemudian ditampilkan melalui LED/buzzer jika memenuhi syarat peringatan, kemudian dikirimkan ke platform IoT untuk disimpan dan divisualisasikan, terakhir data tersebut dianalisis oleh model AI untuk mengklasifikasikan kondisi ruangan.

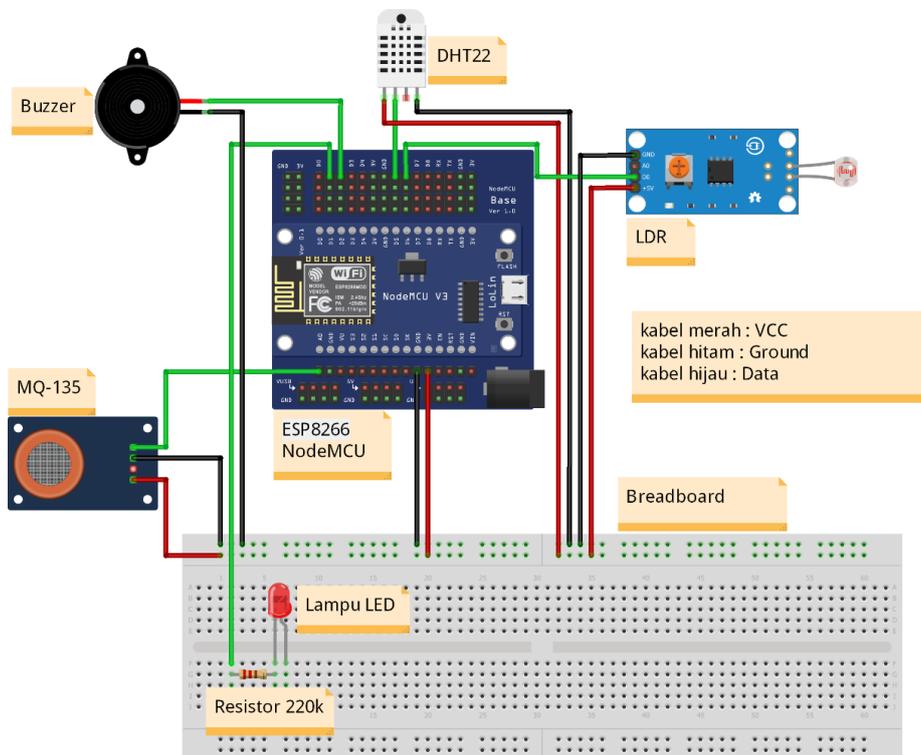
4) Integrasi IoT dan AI

ESP8266 menggunakan koneksi WiFi untuk mengirimkan data sensor ke cloud (ThingSpeak). Setiap data yang terkirim akan disimpan dalam bentuk field dan ditampilkan dalam grafik. Pengguna dapat memantau kondisi ruangan dari mana saja secara online. Model kecerdasan buatan (AI) menggunakan algoritme klasifikasi *decision tree* atau *k-nearest neighbor* yang dilatih menggunakan data dari sensor. AI akan menentukan kondisi ruangan Sehat, Kurang Sehat, atau Bahaya. Keluaran klasifikasi ini bisa digunakan untuk mengaktifkan peringatan otomatis dan dikirim ke dashboard pengguna.

5) Rancangan Skematik Rangkaian

Gambar 2 memperlihatkan diagram pengawatan (*wiring*) antar komponen. NodeMCU ESP8266 berfungsi sebagai otak dari sistem. Mikrokontroler ini menerima *input* dari sensor-sensor, mengirimkan data ke cloud (ThingSpeak) melalui koneksi WiFi, dan mengendalikan *output*, seperti buzzer dan LED.

Sensor DHT22 berfungsi untuk membaca suhu dan kelembaban udara di dalam ruangan. Data dikirim ke pin D5 NodeMCU. Sensor ini memiliki tiga pin: VCC (3.3V), data, dan GND. Sensor MQ-135 adalah sensor gas yang digunakan untuk mendeteksi kualitas udara, seperti CO₂ dan gas berbahaya lainnya. Output analog sensor ini disambungkan ke pin A0 NodeMCU. Sensor LDR (*light dependent resistor*) berfungsi untuk mendeteksi intensitas cahaya. Modul LDR ini menghasilkan output digital (D0) yang dikoneksikan ke pin D0 NodeMCU. Nilai 0 menandakan gelap dan 1 menandakan terang.

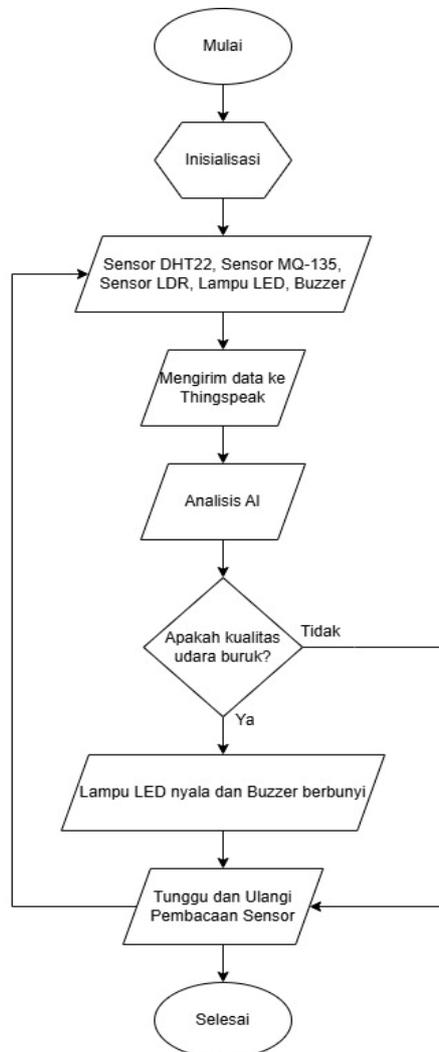


Gambar 2 Skematik rangkaian

LED, *buzzer* aktif, resistor, *breadboard*, dan kabel *jumper*. LED berfungsi sebagai indikator visual yang terhubung ke pin D1 NodeMCU melalui resistor. LED akan menyala saat kondisi lingkungan tidak sehat terdeteksi. *Buzzer* memberikan peringatan suara jika sensor mendeteksi kondisi yang berbahaya, seperti gelap, suhu atau kelembaban tinggi, atau ada gas beracun. *Buzzer* terhubung ke pin D2 NodeMCU. Resistor berfungsi untuk membatasi arus pada LED agar tidak rusak, biasanya menggunakan nilai 220–330 ohm yang terpasang secara seri dengan kaki anoda LED. *Breadboard* digunakan untuk menghubungkan semua komponen secara tidak permanen serta untuk mempermudah penyusunan dan pengujian rangkaian elektronik. Kabel *jumper* digunakan sebagai penghubung antar komponen dan pin pada NodeMCU.

6) Cara Kerja Sistem

Gambar 3 adalah alur kerja sistem *monitoring* lingkungan berbasis IoT dan AI. Proses dimulai saat sistem dinyalakan menggunakan *power supply* melalui kabel USB. Mikrokontroler ESP8266 NodeMCU secara otomatis memulai program utama yang langsung melakukan pembacaan data dari tiga sensor utama, yaitu DHT22 untuk suhu dan kelembaban, MQ-135 untuk kualitas udara, serta LDR untuk pencahayaan ruangan. Setelah data sensor berhasil dibaca, ESP8266 mengirimkan data tersebut ke *cloud platform*, yaitu ThingSpeak, menggunakan koneksi WiFi. Data yang terkirim disimpan dan divisualisasikan pada *dashboard* IoT serta dapat diakses secara *online* oleh pengguna melalui *smartphone* atau laptop.



Gambar 3 Diagram alir kerja sistem

Di sisi *cloud* atau dalam proses lokal, jika AI ditanam di ESP8266 atau server eksternal, maka data sensor yang diterima akan diproses oleh model kecerdasan buatan tersebut. Model ini bertugas melakukan klasifikasi kondisi lingkungan, misalnya apakah ruangan tergolong Sehat, Kurang Sehat, atau Bahaya, berdasarkan nilai suhu, kelembaban, gas, dan pencahayaan. Setelah hasil klasifikasi diperoleh, sistem melakukan pengecekan kondisi. Jika kondisi udara dinyatakan buruk atau membahayakan, maka sistem secara otomatis memberikan notifikasi berupa penyalaaan LED, bunyi *buzzer*, atau pengiriman notifikasi ke perangkat pengguna melalui *platform* IoT. Jika kondisi lingkungan masih tergolong normal, sistem tidak mengaktifkan peringatan. Setelah itu, sistem menunggu selama beberapa detik (sesuai waktu *delay* yang ditentukan) dan kembali membaca ulang sensor untuk memulai proses dari awal lagi. Dengan cara ini, sistem bekerja secara terus-menerus dalam satu siklus *monitoring*, yaitu pengiriman, analisis, dan notifikasi sehingga pengguna dapat mengetahui kondisi udara dalam ruangan secara *real-time* dan otomatis.

7) Desain AI

Dalam penelitian ini, sistem tidak hanya melakukan pemantauan kondisi lingkungan secara *real-time*, namun juga menerapkan logika kecerdasan buatan (AI) untuk melakukan klasifikasi terhadap kondisi udara dalam ruangan. Desain AI bertujuan agar sistem mampu secara otomatis memberikan penilaian terhadap kualitas lingkungan berdasarkan data dari sensor, tanpa perlu pengolahan manual dari pengguna.

Model AI yang digunakan merupakan model klasifikasi sederhana yang mampu membagi kondisi ruangan menjadi beberapa kategori, seperti Sehat, Kurang Sehat, dan Bahaya. Kategori ini didasarkan pada parameter suhu dan kelembaban dari sensor DHT22, kualitas udara (gas) dari sensor MQ-135, dan intensitas cahaya dari sensor LDR

Model klasifikasi dapat menggunakan algoritme seperti *decision tree*, *k-nearest neighbor* (KNN), atau Naive Bayes, tergantung pada kebutuhan dan tingkat kompleksitas. Pada tahap awal, logika klasifikasi bisa diterapkan dengan metode berbasis ambang batas (*threshold logic*). Pada tahap lebih lanjut dapat dikembangkan dengan pelatihan model berbasis data historis. Misalnya, jika suhu lebih besar dari 32°C dan gas tinggi, maka klasifikasinya adalah "Bahaya"; jika suhu normal, tetapi gas sedang, maka klasifikasinya adalah Kurang Sehat; dan jika semua parameter dalam batas aman, maka klasifikasinya adalah Sehat.

8) Pengumpulan Dataset

Dataset dalam sistem ini diperoleh dari pembacaan langsung sensor yang terpasang pada alat. Data dikumpulkan secara berkala dan dikirim ke *cloud platform* (ThingSpeak atau Blynk). Setiap rekaman data berisi nilai suhu (°C), nilai kelembaban (%), nilai gas MQ-135 (ppm), nilai pencahayaan dari LDR (lux atau logika terang/gelap), waktu pencatatan, dan label klasifikasi (jika manual diberi label untuk pelatihan *supervised*).

Data dikumpulkan dalam jumlah yang cukup untuk keperluan analisis dan pelatihan model AI. *Dataset* ini kemudian digunakan untuk melatih model klasifikasi, menguji akurasi AI dalam mengenali pola lingkungan, dan menyusun logika keputusan otomatis dalam sistem.

9) Integrasi AI ke dalam Sistem

AI dapat dijalankan secara lokal di ESP8266 dengan *threshold* sederhana secara eksternal di *cloud* atau *server* menggunakan *platform*, seperti Google Colab, Matlab, atau Python (jika memakai model *machine learning*). *Output* dari AI ini kemudian digunakan untuk mengaktifkan peringatan, seperti LED atau *buzzer*; memberikan informasi status kualitas udara ke pengguna; dan menyimpan klasifikasi di *log database* ThingSpeak.

10) Pembuatan Program

Listing 1 adalah program yang berfungsi sebagai sistem *monitoring* dan peringatan kualitas udara dalam ruangan berbasis mikrokontroler NodeMCU (ESP8266). Setelah perangkat dinyalakan, NodeMCU akan terhubung ke jaringan WiFi menggunakan SSID dan password yang telah ditentukan. Sensor DHT22

Listing 1: Program ESP8266 untuk monitoring suhu, kelembaban, cahaya, dan gas dengan pengiriman data ke ThingSpeak

```
#include <ESP8266WiFi.h>
#include <DHT.h>

#define DHTPIN D5
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

#define LDR_PIN D0 // LDR sebagai input digital
#define LED_PIN D1 // LED indikator
#define BUZZER_PIN D2 // Buzzer
WiFiClient client;
void setup() {
    }
    Serial.println("\nWiFi Terhubung!");
}
void loop() {
    float suhu = dht.readTemperature();
    float kelembaban = dht.readHumidity();
    int gas = analogRead(A0); // Sensor MQ-135
    int cahaya = digitalRead(LDR_PIN); // LDR (0 = terang, 1 = gelap)
    if (isnan(suhu) || isnan(kelembaban)) {
        Serial.println("Gagal membaca DHT22!");
        delay(2000);
        return;
        // LED sebagai indikator bahwa sistem aktif
        digitalWrite(LED_PIN, HIGH);
        // Logika peringatan berdasarkan kondisi sensor
        bool kondisiBahaya = false;

        // Jika gelap
        if (cahaya == 1) {
            Serial.println("⚠ Ruang gelap - Buzzer ON");
            kondisiBahaya = true;
        }
        // Jika suhu > 35 atau kelembaban > 100%
        if (suhu > 35 || kelembaban > 100) {
            Serial.println("⚠ Suhu/kelembaban tinggi - Buzzer ON");
            kondisiBahaya = true;
        }
        // Jika gas berbahaya terdeteksi
        if (gas > 500) {
            Serial.println("⚠ Kebocoran gas terdeteksi - Buzzer ON");
            kondisiBahaya = true;
        }
        // Aktifkan atau matikan buzzer sesuai kondisi
        digitalWrite(BUZZER_PIN, kondisiBahaya ? HIGH : LOW);
        // Kirim data ke ThingSpeak
        if (client.connect(host, 80)) {
            String postStr = "api_key=" + String(apiKey) +
                "&field1=" + String(suhu) +
                "&field2=" + String(kelembaban) +
                "&field3=" + String(gas) +
                "&field4=" + String(cahaya);

            client.stop();
            delay(15000); // Tunggu 15 detik untuk ThingSpeak
        }
    }
}
```

digunakan untuk membaca suhu dan kelembaban udara. Sensor MQ-135 membaca kualitas udara (tingkat gas), sedangkan sensor LDR mendeteksi intensitas cahaya dalam ruangan secara digital (0 untuk terang, 1 untuk gelap). Data dari semua sensor akan ditampilkan pada serial monitor agar pengguna dapat memantau secara lokal.

Program mengevaluasi kondisi lingkungan sebagai berikut: jika ruangan gelap, suhu tinggi (di atas 35°C), kelembaban ekstrem (di atas 100%), atau terdeteksi gas berbahaya (nilai di atas 500), maka *buzzer* akan menyala sebagai peringatan dan LED indikator tetap menyala sebagai tanda sistem aktif. Jika kondisi kembali normal, *buzzer* akan mati otomatis.

Data dari semua sensor akan dikirimkan ke *platform* IoT ThingSpeak setiap 15 detik melalui koneksi HTTP POST. *Platform* ini menampilkan data dalam bentuk grafik yang dapat diakses secara *online* sehingga memungkinkan pemantauan jarak jauh secara *real-time*.

Secara keseluruhan, sistem ini dapat bekerja secara otomatis untuk membaca, menilai, mengirim, dan merespons kondisi lingkungan menggunakan sensor, koneksi WiFi, dan peringatan lokal.

11) Penampilan Data pada ThingSpeak dari ESP8266

Untuk menampilkan data dari sensor ke *platform* ThingSpeak, dilakukan serangkaian tahapan integrasi antara perangkat keras dan perangkat lunak. Mikrokontroler ESP8266 diprogram agar dapat membaca data dari sensor-sensor lingkungan, kemudian mengirimkan data tersebut ke server ThingSpeak melalui koneksi WiFi secara berkala.

Langkah pertama yang dilakukan yaitu membuat akun dan kanal baru pada ThingSpeak. Kanal tersebut diberi nama sesuai kebutuhan sistem *monitoring*, kemudian disesuaikan *field*-nya dengan jumlah parameter yang ingin ditampilkan, seperti suhu, kelembaban, gas, dan cahaya. Setelah kanal dibuat, *API key* yang disediakan oleh ThingSpeak dicatat, khususnya *write API key* yang digunakan dalam proses pengiriman data. Kode program pada Arduino IDE dimodifikasi agar dapat membaca data dari sensor dan mengirimkannya ke ThingSpeak melalui HTTP *request*. Mikrokontroler ESP8266 kemudian disambungkan ke jaringan WiFi menggunakan SSID dan *password* yang telah ditentukan. Setelah koneksi berhasil, mikrokontroler membaca nilai dari sensor DHT22, MQ-135, dan LDR. Nilai-nilai tersebut disusun dalam format yang sesuai dengan standar pengiriman data ThingSpeak dan dikirimkan menggunakan protokol POST.

Secara berkala, data yang dikirimkan oleh ESP8266 muncul dalam bentuk grafik pada masing-masing *field* di ThingSpeak. Grafik ini memungkinkan pengguna memantau kondisi lingkungan secara *real-time* melalui *browser* atau aplikasi. Dengan menerapkan proses ini, sistem berhasil mengirim dan menampilkan data secara kontinu sehingga pengguna dapat melakukan pemantauan lingkungan secara mudah dan terpusat melalui internet.

III. HASIL DAN PEMBAHASAN

Penelitian ini menghasilkan sebuah sistem monitoring dan peringatan kualitas udara berbasis IoT dan kecerdasan buatan yang mampu mendeteksi kondisi lingkungan dalam ruangan secara otomatis. Hasil pengujian dilakukan untuk mengamati kinerja sensor, akurasi klasifikasi AI, serta efektivitas sistem dalam memberikan notifikasi saat kondisi berbahaya terdeteksi.

A. Tampilan Data ThingSpeak dan Analisis Kerja Sensor

Sistem berhasil mengirimkan data suhu, kelembaban, intensitas cahaya, dan kualitas udara dari sensor secara *real time* ke *platform* ThingSpeak. Setiap parameter ditampilkan dalam grafik terpisah yang memudahkan pengguna untuk memantau kondisi lingkungan sepanjang waktu. *Platform* ini juga menyimpan data historis yang berguna untuk analisis tren lingkungan dalam jangka panjang. Setelah sistem

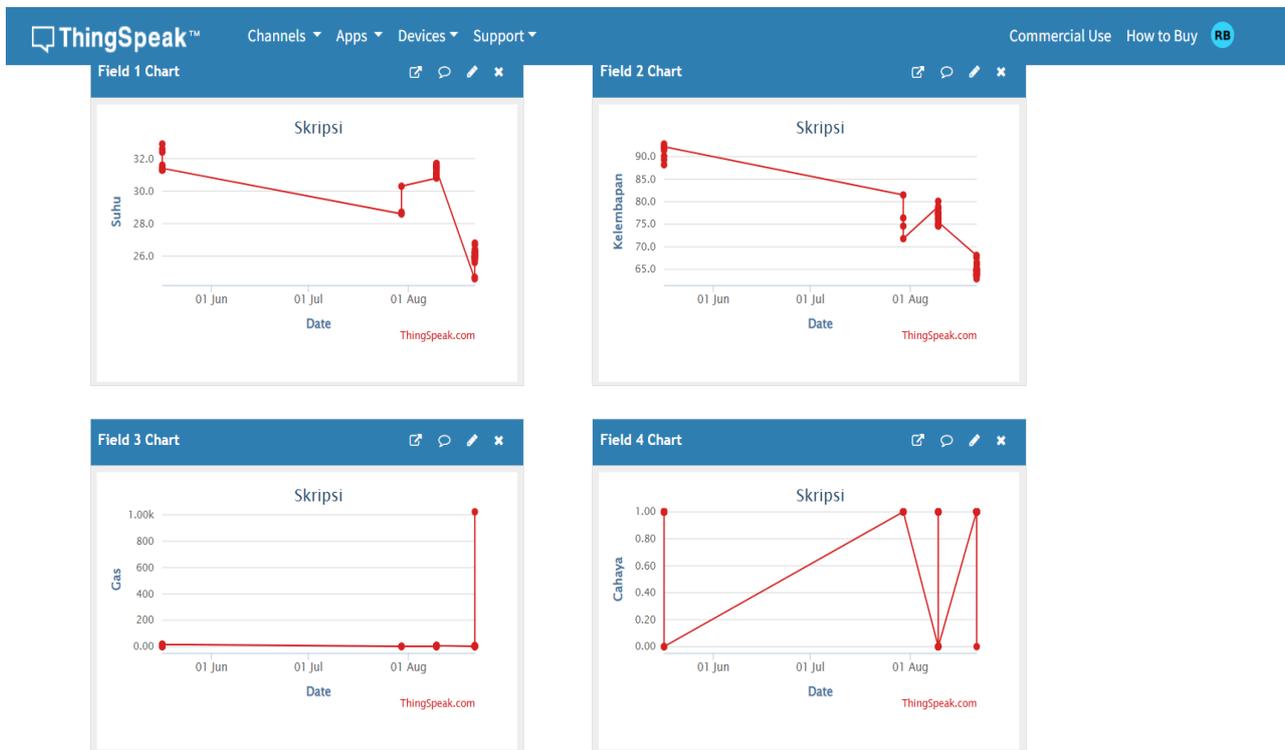
terhubung ke jaringan WiFi dan berhasil membaca data dari sensor DHT22, MQ-135, dan LDR, data dikirim secara berkala ke platform ThingSpeak setiap 15 detik. ThingSpeak menampilkan data tersebut dalam bentuk grafik *real-time* pada masing masing *field*, yaitu: *field-1* untuk suhu (dalam °C) dari sensor DHT22; *field-2* untuk kelembaban udara (dalam %) dari sensor DHT22; *field-3* untuk nilai gas (polusi udara) dari sensor MQ-135; dan *field-4* untuk intensitas cahaya dari sensor LDR (nilai 1 untuk terang, 0 untuk gelap).

Gambar 4 menunjukkan tampilan data dalam bentuk grafik yang diperbarui sesuai interval pengiriman. Grafik ini memudahkan pengguna untuk memantau kondisi lingkungan secara visual dan historis. Melalui tampilan ThingSpeak, pengguna dapat mengetahui tren perubahan suhu, kelembaban, gas, dan cahaya di dalam ruangan selama periode tertentu. Hal ini sangat bermanfaat untuk menganalisis pola lingkungan dan mengambil tindakan preventif bila terjadi kondisi berbahaya. Contoh tampilan data grafik pada ThingSpeak yang ditampilkan dalam masing-masing *field*:

- *field-1* (suhu) menunjukkan fluktuasi suhu ruangan dalam rentang waktu tertentu.
- *field -2* (kelembaban) menggambarkan tingkat kelembaban relatif.
- *field -3* (gas) merepresentasikan tingkat kualitas udara (semakin tinggi nilai, semakin tercemar).
- *field -4* (cahaya): menunjukkan status pencahayaan ruangan secara biner.

Dengan fitur penyimpanan dan visualisasi otomatis ini, ThingSpeak memudahkan pengguna dalam melakukan pemantauan tanpa perlu menyimpan data secara manual.

Sensor DHT22 mampu membaca suhu dan kelembaban dengan baik dalam kisaran suhu 25°C hingga 38°C dan kelembaban 50% hingga 85%. Sensor MQ-135 memberikan nilai analog untuk kualitas udara, di mana nilai lebih dari 500 dianggap sebagai indikasi adanya gas berbahaya. Sensor LDR merespons dengan baik terhadap perubahan pencahayaan dengan *output* digital (1 untuk terang, 0 untuk gelap). *Buzzer* aktif saat kondisi gelap, kelembaban tinggi, suhu tinggi, atau polusi udara terdeteksi.



Gambar 4 Tampilan data

B. Analisis Klasifikasi Kecerdasan Buatan dan Uji Sistem

Data dari sensor digunakan untuk melatih dan menguji algoritme klasifikasi berbasis *decision tree* dan *k-nearest neighbor* (KNN) menggunakan platform Python. Dataset yang digunakan terdiri dari variabel suhu, kelembaban, gas, dan cahaya. Algoritma klasifikasi menghasilkan prediksi kondisi lingkungan Sehat, Kurang Sehat, atau Bahaya. Berdasarkan uji validasi, akurasi yang dicapai oleh *decision tree* adalah 85,7%, sedangkan KNN mencapai 83,3%.

Pengujian sistem dilakukan dengan melakukan variasi kondisi lingkungan, seperti: ruangan terang dengan suhu normal dan udara bersih, maka sistem tidak memberikan notifikasi sehingga LED dan *buzzer* tetap *off*; ruangan gelap (sensor LDR = 1), maka sistem mendeteksi gelap sehingga *buzzer* dan LED aktif; suhu di atas 35°C, maka sensor DHT22 mendeteksi suhu tinggi sehingga sistem mengaktifkan LED dan *buzzer*; dan gas tinggi (MQ-135 di atas 500), maka sistem menganggap kondisi berbahaya sehingga LED dan *buzzer* aktif.

C. Evaluasi Respon Sistem dan Hasil Pengamatan

Waktu respon dari pembacaan sensor hingga pengiriman data ke ThingSpeak dan pemberian *output* notifikasi lokal, rata-rata memakan waktu sekitar 1–2 detik. Komponen *output* (LED dan *buzzer*) menyala secara cepat saat ambang batas dilampaui. Data juga berhasil dikirim ke ThingSpeak secara konsisten setiap 15 detik sesuai *delay* yang diatur. Tidak ditemukan keterlambatan sistem selama WiFi stabil dan daya mencukupi. Namun, pada kondisi sinyal WiFi lemah, pengiriman data ke *cloud* dapat mengalami gangguan sehingga perlu dipastikan koneksi internet dalam keadaan baik.

Sistem berhasil memberikan respon notifikasi melalui *buzzer* secara cepat saat terdeteksi kondisi lingkungan tidak sehat. Respon ditunjukkan saat pencahayaan ruangan rendah (gelap), suhu melebihi 35°C atau kelembaban melebihi 75%, dan nilai gas melebihi ambang batas (500).

Hasil uji coba alat berdasarkan kondisi lingkungan tertentu pada Tabel I menunjukkan bahwa sistem telah mampu mendeteksi kondisi lingkungan secara otomatis dan memberikan peringatan dengan baik.

TABEL I
HASIL UJI COBA ALAT

No.	Nama Sensor	Kondisi Pengaktifan	Output	Nilai Yang Diukur
1	DHT22	suhu > 35°C	<i>buzzer</i> aktif	37°C
2	DHT22	kelembaban > 85%	<i>buzzer</i> aktif	88%
3	MQ-135	nilai gas > 500 (ADC)	<i>buzzer</i> aktif	650
4	LDR	nilai = 0 (gelap)	<i>buzzer</i> aktif	0 (gelap)
5	Semua normal	suhu < 35°C, gas < 400, terang	Tidak aktif	Sesuai batas aman

IV. SIMPULAN

Penelitian ini berhasil merancang dan mengimplementasikan sistem peringatan kualitas udara berbasis *internet of things* (IoT) dan kecerdasan buatan (AI) dengan memanfaatkan mikrokontroler ESP8266 NodeMCU. Sistem ini menggunakan tiga jenis sensor, yaitu DHT22 untuk mengukur suhu dan kelembaban, MQ-135 untuk mendeteksi gas berbahaya, dan LDR untuk mengukur intensitas cahaya dalam ruangan. Seluruh data yang dikumpulkan dikirimkan ke *cloud platform* ThingSpeak secara *real-time* dan divisualisasikan dalam bentuk grafik.

Sistem ini dapat menampilkan informasi maupun peringatan ketika kondisi lingkungan dianggap tidak sehat, seperti suhu tinggi, kelembaban tinggi, pencahayaan rendah, atau terdeteksi gas berbahaya. Notifikasi diberikan melalui *buzzer* dan LED sebagai peringatan lokal. Penggunaan algoritme *decision tree* dan *k-nearest neighbors* (KNN) mampu mengklasifikasikan kondisi lingkungan menjadi kategori Sehat, Kurang Sehat, dan Bahaya dengan akurasi yang cukup baik.

Berdasarkan hasil pengujian, sistem mampu merespon perubahan lingkungan secara cepat dan akurat. Keandalan dalam pengiriman data ke *cloud* dan pemrosesan lokal menjadikan alat ini layak digunakan sebagai sistem *monitoring* dan peringatan kualitas udara dalam ruangan.

Sistem ini juga memiliki potensi untuk dikembangkan lebih lanjut, baik dari sisi sensor tambahan, integrasi *mobile app*, maupun kemampuan otomatisasi tindakan berdasarkan hasil klasifikasi. Penelitian ini baru sebatas pada penggunaan sensor MQ-135 (gas), DHT22 (suhu & kelembaban), dan LDR (cahaya). Padahal indikator kualitas udara yang umum dipakai secara internasional mencakup PM2.5, PM10, CO, NO₂, SO₂, dan O₃ sehingga masih membutuhkan tambahan sensor partikel debu (PM2.5 atau PM10) dan sensor gas spesifik (CO, NO₂, dan SO₂). Hal ini akan membuat sistem lebih representatif terhadap standar kualitas udara menurut WHO. Untuk itu masih terbuka peluang pengembangan dari penelitian ini supaya lebih informatif dan lengkap.

DAFTAR REFERENSI

- [1] A. R. A. Haruna, Lahming, and F. Amir, "Pencemaran udara akibat gas buang kendaraan bermotor," *UNM Environ. J.*, vol. 2, no. 2, pp. 57–61, 2019, doi: 10.26858/uej.v2i2.10092.
- [2] Y. Sesempuli, B. Iswanto, and D. I. Hendrawan, "Kajian tingkat kerusakan pencemaran di Sungai Krukut," *J. Tek. Lingkung.*, vol. 26, no. 2, pp. 11–30, 2020, doi: 10.5614/j.tl.2020.26.1.2.
- [3] R. Umah and E. Gusmira, "Dampak pencemaran udara terhadap kesehatan masyarakat di perkotaan," *Profit J. Manaj., Bisnis Akunt.*, vol. 3, no. 3, pp. 103–112, 2024, doi: 10.58192/profit.v3i3.2246.
- [4] M. S. M. N. Kusuma, "Pemodelan polusi udara akibat pengalihan lalu lintas dari pembangunan fly over Aloha Sidoarjo," *J. Ilmu Lingkung.*, vol. 22, no. 4, pp. 923–932, 2024, doi: 10.14710/jil.22.4.923-932.
- [5] R. Wangintan and A. Sofyan, "Analisis kualitas udara ambien dan penentuan lokasi stasiun pemantauan kualitas udara (SPKU) DKI Jakarta," *J. Teknol.*, vol. 8, no. 1, pp. 31–35, 2019.
- [6] G. M. Maula, "Efektivitas implementasi kebijakan pengendalian pencemaran udara di Indonesia," *Savana Indones. J. Nat. Resour. Environ. Law*, vol. 1, no. 2, pp. 145–159, 2024, doi: 10.25134/savana.v1i2.223.
- [7] A. Pradifan, W. Widayat, and A. Suprihanto, "Pemantauan kualitas udara Kota Tegal (studi kasus: Kecamatan Tegal Selatan, Kecamatan Tegal Barat, Kecamatan Tegal Timur)," *J. Ilmu Lingkung.*, vol. 19, no. 1, pp. 73–82, 2021, doi: 10.14710/jil.19.1.73-82.
- [8] M. Hasanuddin and H. Herdianto, "Sistem monitoring dan deteksi dini pencemaran udara berbasis Internet of Things (IoT)," *J. Comput. Syst. Informatics*, vol. 4, no. 4, pp. 976–984, 2023, doi: 10.47065/josyc.v4i4.4034.
- [9] N. B. M. N. Riswandha and R. Selviana, "Analisis pencemaran udara terhadap kualitas hidup masyarakat," *J. Spirit*, vol. 16, no. 2, pp. 122–130, 2023, doi: 10.53567/spirit.v16i2.363.
- [10] M. R. A. Akbar, E. Priatna, Sutisna, and I. Taufiqurohman, "Monitoring kualitas udara menggunakan NodeMCU ESP8266 berbasis Internet of Things (IoT) di Ciamis," *E-Joint Electron. Electr. J. Innov. Technol.*, vol. 3, no. 2, pp. 73–78, 2022, doi: 10.35970/e-joint.v3i2.1687.
- [11] D. P. Sari, "Prototype alat monitoring suhu, kelembaban dan kecepatan angin untuk smart farming menggunakan komunikasi LoRa dengan daya listrik panel surya," *Kilat*, vol. 10, no. 2, pp. 370–380, 2021, doi: 10.33322/kilat.v10i2.1376.
- [12] A. D. Ramadhani, N. Ningsih, A. Nurcahya, and N. Azizah, "Klasifikasi dan monitoring kualitas udara dalam ruangan menggunakan ThingSpeak," *J. Tek. Elektro Komput. Triac*, vol. 10, no. 1, pp. 1–5, 2023, doi: 10.21107/triac.v10i1.17501.
- [13] R. Muttaqin, W. S. W. Prayitno, N. E. Setyaningsih, and U. Nurbaiti, "Rancang bangun sistem pemantauan kualitas udara berbasis IoT dengan sensor DHT11 dan MQ135," *J. Pengelolaan Lab. Pendidik.*, vol. 6, no. 2, pp. 102–115, 2024, doi: 10.14710/jplp.6.2.102-115.
- [14] K. Exaudi et al., "Purwarupa sistem pemantauan polusi udara di ruang tertutup menggunakan platform ThingSpeak," *JTT J. Teknol. Ter.*, vol. 9, no. 2, p. 101, 2023, doi: 10.31884/jtt.v9i2.412.
- [15] E. Soesilo and I. Fayuza, "Prototype alat pendeteksi kebocoran gas LPG berbasis IoT," *J. Tek. Ind. Terintegrasi*, vol. 6, no. 3, pp. 862–879, 2023, doi: 10.31004/jutin.v6i3.17664.

- [16] Y. Tjandi, P. Studi, T. Elektro, and U. N. Makassar, "Gas sensor tools in IoT-based office," *Metrik*, vol. 20, no. 2, 2023, doi: 10.59562/metrik.v20i2.5535.
- [17] A. S. Sumaedi, F. R. R. Rosman, and F. F. Fiqri, "Perancangan sistem keamanan pendeteksi gas dalam ruangan menggunakan sensor MQ-2 berbasis mikrokontroler Arduino Uno R3," *J. Siskom-KB Sist. Komput. Kecerdasan Buatan*, vol. 7, no. 3, pp. 198–207, 2024, doi: 10.47970/siskom-kb.v7i3.675.
- [18] P. D. Nugraha, R. Soekarta, and I. Amri, "Rancang bangun alat monitoring suhu dan kelembaban berbasis IoT pada gudang obat RS Aryoko Sorong," *Framework*, vol. 2, no. 1, pp. 21–31, 2023. [Online]. Available: <https://ejournal.um-sorong.ac.id/index.php/jiki/article/download/3044/1636/12421>
- [19] A. Pariama and R. Efendi, "Pengembangan sistem deteksi kualitas udara berbasis sensor IoT menggunakan fitur notifikasi Telegram di Terminal Tingkir, Salatiga," *J. JIMIK*, vol. 6, no. 2, pp. 1025–1034, 2025, doi: 10.63447/jimik.v6i2.1384.
- [20] Y. Wahyuni, S. Febiola, and K. Kunci, "Alat pendeteksi kadar udara dan asap lingkungan sekitar berbasis Android," *Semtik*, vol. 4, no. 1, pp. 150–157, 2025, doi: 10.31284/p.semtik.2025-1.7173.

Rangga Bimo Hermawan, mahasiswa program sarjana Sistem Komputer Universitas Gunadarma Jakarta. Bidang yang ditekuni adalah sistem kendali. Saat ini sudah lulus dan memperoleh gelar Sarjana Komputer (S.Kom.)

Setiyono, memperoleh gelar Sarjana Teknik Elektro di Universitas Gunadarma, gelar Magister Teknik Elektro Universitas Gunadarma, dan gelar Doktor di Universitas Gunadarma Jakarta. Bidang yang ditekuni elektronika daya dan teknik tenaga listrik. Saat ini sebagai pengajar di Jurusan Teknik Elektro Universitas Gunadarma.