



RANCANG BANGUN SISTEM DETEKSI ASAP ROKOK DAN DOKUMENTASI MAHASISWA BERBASIS IOT DENGAN INTEGRASI TELEGRAM DAN ESP32-CAM

Abdul Hafid Hidayat^{*1}, Diah Rahmawati², Nia Komalasari³

^{1,2,3} Teknik Informatika, Fakultas Teknik, Universitas Islam Syekh-Yusuf

Jl. Maulana Yusuf, Babakan, Kota Tangerang, Banten, 15118

*e-mail: ¹ 2104030040@students.unis.ac.id, ² drahmawati@unis.ac.id, ³ nia@unis.ac.id

Abstrak

Penelitian ini merancang dan membangun sistem deteksi asap rokok berbasis Internet of Things (IoT) yang terintegrasi dengan ESP32-CAM dan bot Telegram. Sistem dirancang untuk mendeteksi asap rokok di lingkungan kampus, khususnya area yang rawan pelanggaran, serta melakukan dokumentasi otomatis berupa pengambilan gambar saat asap terdeteksi. Sensor MQ-2 digunakan untuk mengukur konsentrasi asap, kemudian mikrokontroler ESP32 memicu ESP32-CAM mengambil foto dan mengirim notifikasi secara real-time ke Telegram. Pengujian menggunakan metode Black Box Testing menunjukkan sistem mampu mendeteksi asap dengan ambang batas tertentu dan mengirimkan informasi kepada pihak terkait dengan cepat dan akurat. Hasil penelitian ini diharapkan dapat mendukung upaya pengawasan larangan merokok di lingkungan kampus dan menjadi referensi untuk pengembangan sistem serupa.

Kata Kunci: IoT, ESP32-CAM, MQ-2, Telegram, deteksi asap rokok, dokumentasi.

Article History

Received: Agustus 2025

Reviewed: Agustus 2025

Published: Agustus 2025

Plagiarism Checker No 712

Prefix DOI : Prefix DOI :

10.8734/Kohesi.v1i2.365

Copyright : Author

Publish by : Kohesi



This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/)

Pendahuluan

Gas beracun ditimbulkan oleh rokok yaitu salah satu bentuk dari penyebab polusi udara (Tendra & Wulandari, 2020). Asap rokok yang dikeluarkan oleh perokok berdampak negatif pada kesehatan orang yang terkena asapnya, yang dikenal sebagai perokok aktif. Meski demikian, dampak terburuk tetap dirasakan oleh perokok. Selain menghirup asap rokok langsung, perokok aktif juga berisiko besar menghirup kembali asap yang dikeluarkannya. Asap yang keluar dari ujung rokok (bagian yang terbakar) dikenal sebagai asap sampingan. Penelitian menunjukkan bahwa asap sampingan mengandung lebih banyak zat berbahaya hasil pembakaran tembakau dibandingkan asap utama (Basuki & Efendi, 2022). Saat ini masih banyak orang yang tidak mematuhi peraturan dilarang merokok pada kawasan tertentu salah satunya seperti di lembaga pendidikan. Paparan dari asap rokok tidak hanya berdampak pada perokok aktif, tetapi juga pada perokok pasif yang terpaksa menghirup asap tersebut. Hal ini menjadi perhatian serius, terutama di tempat umum dan institusi pendidikan, di mana anak remaja berisiko tinggi terpapar.

Lingkungan institusi pendidikan, terutama di area yang seharusnya bebas asap rokok. Asap rokok tidak hanya mengganggu kenyamanan tetapi juga berdampak buruk terhadap kesehatan, khususnya bagi mahasiswa dan tenaga pengajar. Untuk meminimalkan bahaya asap rokok, sejumlah langkah diambil, termasuk pemasangan stiker/spanduk larangan merokok. Akan tetapi, cara ini kurang berhasil karena masih ada individu yang belum menyadari pentingnya larangan merokok di tempat umum dan khusus (Anwari et al., 2023).

Dalam Peraturan Gubernur Nomor 40 Tahun 2020 menjelaskan pedoman mengenai pembinaan, pengawasan, dan penegakan hukum terkait kawasan dilarang merokok, termasuk di tempat umum dan institusi pendidikan (Governor of Special Capital Territory of Jakarta,



2020). Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk membuat sistem pendeteksi asap berbasis IoT, teknologi IoT dapat dimanfaatkan secara maksimal untuk meningkatkan keamanan dan mendukung ketertiban kondisi dilingkungan institusi pendidikan.

Dalam satu dekade ini, komputer dan manusia hampir sepenuhnya tergantung pada internet untuk segala informasi. Membicarakan Internet of Things (IoT) tidak akan ada habisnya karena IoT dapat diterapkan pada keseharian kita dengan benda-benda yang dapat dijadikan perangkat untuk mempermudah aktivitas sehari-hari (Muafani, 2020). Internet of Things (IoT) yaitu ide dimana konektifitas internet dapat bertukar informasi satu sama lainnya dengan benda yang ada disekelilingnya. Pengertian Internet of Things menurut Kevin Ashton dalam e-book yang berjudul "Making Sense of IoT" "Pengertian 'Internet of Things' adalah sensor-sensor yang terhubung ke internet dan berperilaku seperti internet dengan membuat koneksi-koneksi terbuka setiap saat, serta berbagi data secara bebas dan memungkinkan aplikasi-aplikasi yang tidak terduga, sehingga komputer dapat memahami dunia sekitar mereka menjadi bagian dari kehidupan manusia" (Kakihary, 2021). Dalam beberapa tahun terakhir, teknologi Internet of Things (IoT) telah menunjukkan potensi besar dalam menciptakan solusi untuk berbagai masalah, termasuk pada sektor pendidikan dan kesehatan. IoT memungkinkan alat-alat yang saling berkomunikasi dan berbagi data secara real-time, memberikan peluang untuk membangun sistem monitoring dan kontrol yang lebih efisien.

Pengembangan sistem yang dapat mendeteksi asap rokok sekaligus mendokumentasi mahasiswa secara otomatis menjadi kebutuhan mendesak dengan memanfaatkan Base Plate ESP32 Devkit V1, ESP32 Devkit V1, Buzzer, FTDI FT232RL, MQ-2 dan ESP32-CAM. Sistem ini dapat melakukan dokumentasi mahasiswa serta mendeteksi pelanggaran berupa asap rokok secara real-time. Platform Telegram akan diintegrasikan untuk memberikan akses monitoring jarak jauh kepada pengelola sistem. Penelitian ini dilakukan di tangga darurat lantai 3 (tiga) Fakultas Teknik Universitas Islam Syekh-Yusuf. Lokasi ini dipilih karena merupakan area yang sering dijadikan tempat untuk mahasiswa merokok, sehingga ideal untuk menguji efektivitas sistem dalam mendeteksi asap rokok secara real-time. Dalam pengembangan sistem ini, peneliti menggunakan metode prototype. Metode ini dipilih karena memungkinkan iterasi dan umpan balik yang cepat dari pengguna yaitu wakil dekan 1 (satu) selaku bidang akademik dan dakwah, sehingga sistem dapat disesuaikan dengan kebutuhan pengguna secara lebih efektif.

Landasan Teori

Internet of things yang sering disebut dengan IoT adalah sistem yang dapat memantau dan menggerakkan perangkat dari jarak jauh menggunakan teknologi komunikasi internet (Aulia et al., 2024). Internet of Things merupakan ide di mana berbagai objek fisik, seperti perangkat elektronik, kendaraan, peralatan rumah tangga, dan sebagainya, terkoneksi dan berkomunikasi melalui jaringan internet. Dalam konsep IoT, objek-objek tersebut dilengkapi dengan sensor, perangkat lunak, dan koneksi internet yang memungkinkan mereka mengumpulkan, bertukar, dan mengirim data secara otomatis. Prinsip dasar di balik IoT adalah memberdayakan objek-objek tersebut untuk menghimpun informasi dari sekitarnya dan berbagi data melalui internet. Melalui konektivitas ini, objek-objek dapat berinteraksi dan bekerjasama tanpa perlu campur tangan manusia secara langsung (Riadi et al., 2024).

Dalam konsep IoT, setiap objek atau perangkat yang terhubung memiliki identitas unik dan kemampuan untuk mentransfer data melalui jaringan. Perangkat IoT dapat berupa perangkat elektronik konvensional yang dilengkapi dengan sensor dan konektivitas internet, seperti smartphone, kendaraan, perangkat rumah pintar, atau peralatan industri. Mereka juga dapat berupa sensor atau perangkat khusus yang didesain secara khusus untuk tujuan tertentu, seperti sensor suhu, detektor gerakan, atau sistem pemantauan lingkungan (Sohail Aslam¹, Maqsood Ahmad², 2021).

Metode

A. Tempat dan Waktu Penelitian

Fakultas Teknik Universitas Islam Syekh-Yusuf yang berlokasi di Jalan Maulana Yusuf No.10, Babakan, Kecamatan Tangerang, Babakan, Kec. Tangerang, Kota Tangerang, Provinsi Banten (15118).

B. Observasi

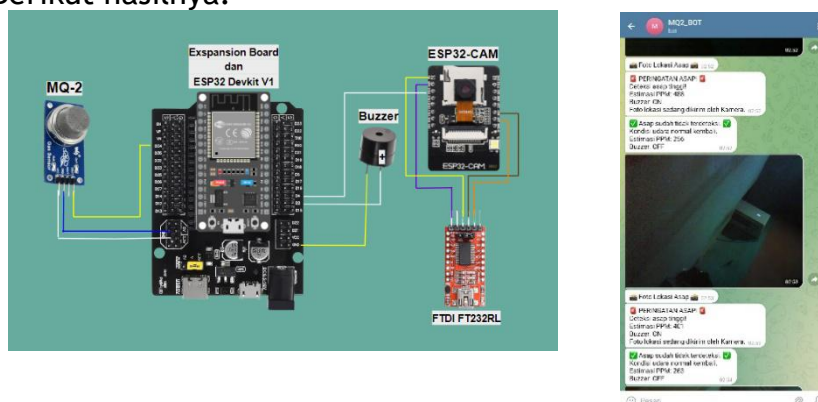
Pengamatan dilakukan langsung pada lantai 3 (Tiga) tangga darurat Fakultas Teknik Universitas Islam Syekh-Yusuf. Observasi berlangsung kurang lebih 2 bulan, dimulai pada bulan Januari 2025 sampai bulan Februari 2025. Peneliti melakukan pengamatan langsung secara sistematis terkait masalah yang diteliti.

C. Wawancara /Listen To Customer

Peneliti mengumpulkan informasi mengenai permasalahan yang dihadapi. Wawancara dilakukan dengan Wakil dekan pada tanggal 24 Februari 2025. Hasil wawancara tersebut bahwa tantangan terbesar yang dihadapi adalah banyaknya mahasiswa yang mengabaikan larangan merokok dilingkungan kampus tepatnya ditangga darurat lantai 3 (tiga). Serta diharapkan dengan dibuatkannya alat pendeteksi asap rokok dan dokumentasi mahasiswa dengan integrasi telegram dan ESP32-CAM, agar mahasiswa akan lebih sadar bahwa ada pengawasan serta mahasiswa mungkin akan berfikir 2 (dua) kali sebelum merokok ditempat terlarang.

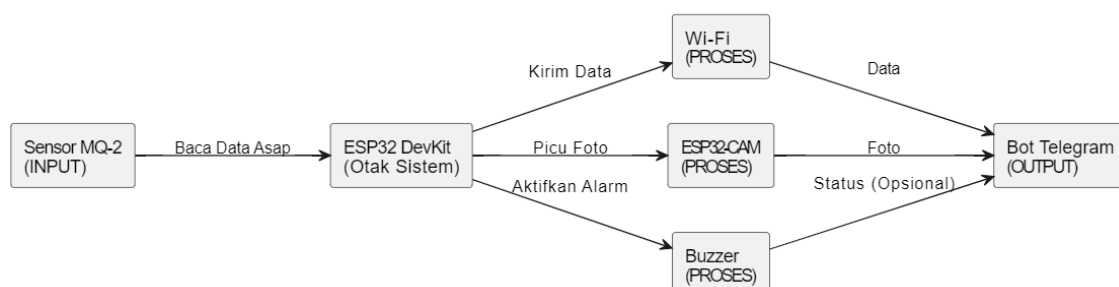
D. Build or Revise Mockup

Desain ini merupakan hasil riset yang dilakukan antara pengembang dan pengguna. Berikut hasilnya:



E. Pengujian Sistem / Test Drive Mockup

Prototype alat yang telah dibuat akan diuji coba dan dievaluasi untuk memastikan bahwa alat tersebut memenuhi harapan yang diinginkan. Apabila hasil pengujian prototipe belum memenuhi spesifikasi yang diinginkan pengguna, pengembang akan melakukan perbaikan hingga alat tersebut memenuhi kriteria yang diinginkan. Berikut adalah tahapan pengujian yang akan dilakukan dalam bentuk diagram blok:





1. Memastikan Sensor Gas MQ-2 memiliki kemampuan mendeteksi keberadaan asap rokok dengan mengukur konsentrasi zat kimia di udara. Ketika kadar gas meningkat melewati ambang batas 300 PPM, sensor akan memberikan sinyal ke ESP32 DevKit sebagai pengendali utama.
2. Melakukan uji coba dengan sumber asap nyata (misalnya asap rokok) untuk memverifikasi bahwa sensor MQ-2 mendeteksi dan mengirimkan sinyal ke ESP32 DevKit secara akurat. Nilai PPM yang terbaca harus sesuai dengan kondisi sebenarnya.
3. Verifikasi ESP32 DevKit sebagai otak sistem: DevKit membaca data dari sensor MQ-2, memprosesnya, dan memutuskan apakah akan mengaktifkan alarm (Buzzer) serta memicu pengambilan gambar pada ESP32-CAM.
4. Pastikan Buzzer aktif saat terdeteksi asap rokok. Buzzer harus berbunyi atau berkedip sesuai program sebagai tanda peringatan lokal.
5. Verifikasi ESP32-CAM dapat mengambil gambar dengan jelas setelah menerima sinyal pemicu dari ESP32 DevKit.
6. Pastikan kualitas gambar yang diambil oleh ESP32-CAM cukup baik untuk dokumentasi dan pengawasan.
7. Periksa alur pengiriman gambar dari ESP32-CAM ke ESP32 DevKit (jika ada intermediate processing) atau langsung ke Bot Telegram melalui koneksi Wi-Fi.
8. Verifikasi ESP32 DevKit dan ESP32-CAM sama-sama terhubung ke jaringan Wi-Fi yang telah dikonfigurasi dengan stabil, tanpa gangguan koneksi.
9. Pastikan koneksi Wi-Fi stabil selama proses pengiriman data sensor, foto, dan status alarm ke server atau Bot Telegram.
10. Verifikasi Bot Telegram dapat menerima gambar dan pesan notifikasi yang dikirimkan oleh sistem (ESP32 DevKit & ESP32-CAM).
11. Periksa kualitas gambar di Telegram agar gambar yang diterima jelas dan dapat diidentifikasi oleh pengguna.
12. Pastikan Bot Telegram mengirimkan pesan notifikasi dengan informasi relevan, seperti waktu pengambilan gambar, status Buzzer, dan peringatan asap rokok.
13. Uji waktu respons sistem untuk mengukur lamanya proses mulai dari deteksi asap oleh MQ-2, aktivasi Buzzer, pemicu pengambilan gambar oleh ESP32 DevKit, hingga notifikasi diterima di Telegram.
14. Pastikan tidak ada penundaan signifikan yang dapat mengurangi kecepatan respons sistem, sehingga deteksi dan peringatan dapat dilakukan secara real-time.

F. Analisa sistem yang berjalan saat ini

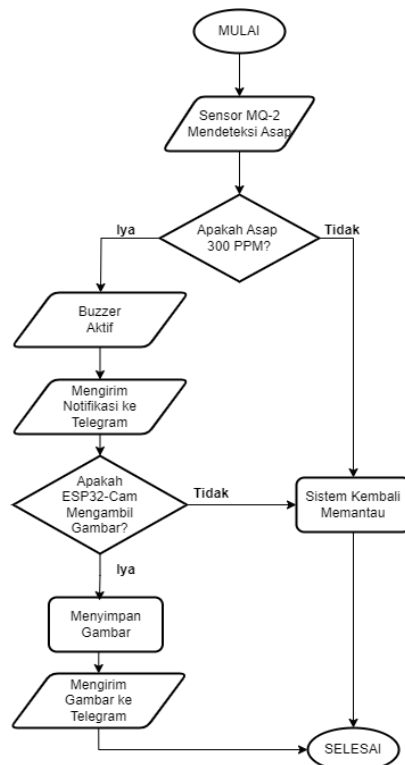
Setelah melakukan observasi dan wawancara berikut sistem yang berjalan saat ini, sistem akan dirancang menggunakan diagram alir (*flowchart*):



1. Mulai, dimulai ketika seseorang melewati area Fakultas Teknik, khususnya ditangga darurat lantai 3.
2. Seseorang (biasanya *office boy* atau dosen) melewati area tersebut atau kebetulan melintas ditangga darurat lantai 3.
3. Orang tersebut mengamati langsung kondisi sekitar.
4. Keputusan:
 - a. Jika ada mahasiswa yang merokok, orang yang menemukan pelanggaran akan menegur mahasiswa tersebut secara langsung, meminta untuk mematikan rokok, dan mengingatkan peraturan kampus.
 - b. Jika tidak ada mahasiswa yang merokok, Orang tersebut melanjutkan aktivitas seperti biasa, tanpa melakukan tindakan apa pun.
5. Selesai

G. Analisa perancangan yang diusulkan

Setelah mengumpulkan seluruh kebutuhan sistem, tahap berikutnya adalah merancang prototipe untuk sistem yang diajukan oleh pengguna. Berikut ini adalah flowchart rancangan yang diusulkan.



1. Sistem mulai bekerja dan dalam kondisi aktif.
2. Perangkat pendeteksi gas tipe MQ-2 mampu mengenali keberadaan asap, membaca kualitas udara dan mengecek apakah ada asap rokok.
3. Apakah Asap Melebihi Ambang Batas?
4. Jika YA → Buzzer aktif
5. Mengirim notifikasi ke telegram dan memicu untuk ESP32-Cam mengambil gambar
6. Jika TIDAK → Sistem kembali ke pemantauan tanpa melakukan tindakan lebih lanjut
7. Lanjut ke langkah berikutnya (Apakah ESP32-CAM mengambil gambar?)
8. Jika YA → ESP32-CAM langsung menangkap gambar lokasi
9. Mengirim gambar ke telegram → gambar yang diambil akan langsung dikirimkan ke bot telegram sebagai notifikasi *real-time*.
10. Jika TIDAK → Sistem kembali ke pemantauan tanpa melakukan tindakan lebih lanjut
11. Selesai.

Temuan dan Pembahasan

Pada tahapan ini dilaksanakan dengan mengacu pada metode prototyping. Pada tahap perencanaan sistem, proses mencakup identifikasi kebutuhan pengguna, serta evaluasi terhadap prototipe yang dikembangkan. Dalam tahap ini juga dijabarkan mengenai strategi perancangan, serta analisis komponen dan fungsi yang dibutuhkan untuk merealisasikan alat yang akan dirancang:

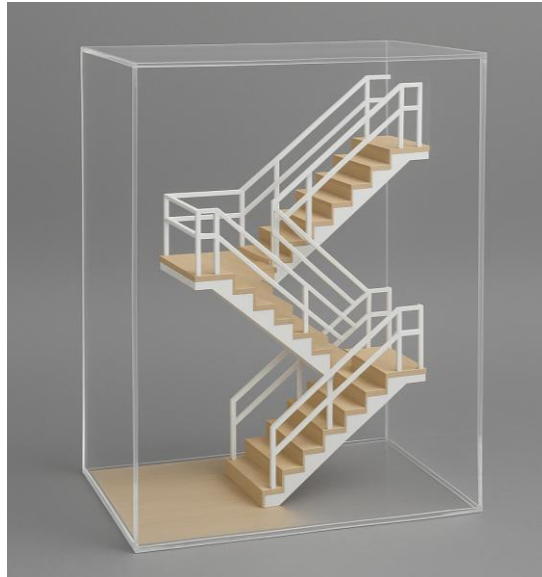
A. Listen to Customer

Listen to Customer atau identifikasi kebutuhan pengguna merupakan tahapan awal, bertujuan untuk memperoleh informasi secara langsung dari pengguna terkait kebutuhan dan ekspektasi terhadap sistem yang akan dikembangkan. Melalui tahapan ini, peneliti dapat memahami secara lebih mendalam karakteristik permasalahan di lapangan serta merumuskan solusi yang relevan dan tepat sasaran.

Untuk menguji kebenaran hipotesis secara empiris, proses pengumpulan data dilakukan secara menyeluruh dan mendalam. Dengan demikian, tahapan ini menjadi fondasi penting bahwa sistem yang dirancang benar-benar menjawab kebutuhan pengguna.

B. Build or Revise Mock-up

Dalam tahap ini merupakan tahapan perancangan berdasarkan hasil analisis kebutuhan. Pada fase ini, dilakukan perancangan awal alat dengan mempertimbangkan aspek teknologi dan karakteristik pengguna. Langkah awal dimulai dengan identifikasi kebutuhan sistem, yang mencakup penentuan spesifikasi masukan (*input*), keluaran (*output*), serta proses yang dibutuhkan untuk menghasilkan *output* yang sesuai dengan tujuan sistem.



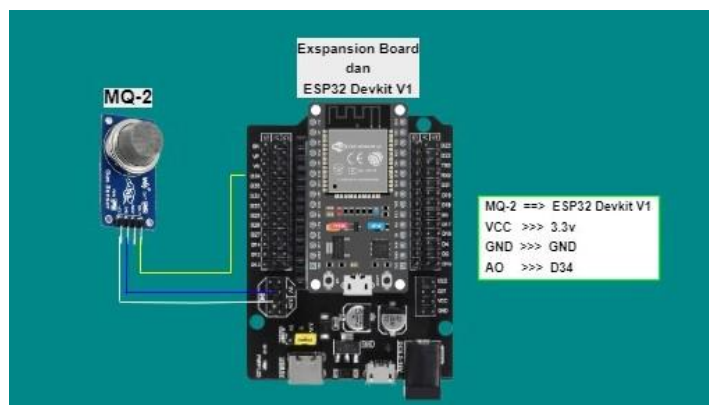
Untuk menentukan posisi sensor yang tepat, model ini dirancang dengan mempertimbangkan arah ventilasi dan pergerakan asap. Ketika konsentrasi asap terdeteksi melebihi ambang batas aman oleh sensor MQ-2, sistem akan mengaktifkan peringatan berupa LED dan buzzer. Sistem ini dibangun menggunakan ESP32 Devkit V1 yang dipasang pada Expansion Board ESP32 Devkit V1, serta didukung oleh ESP32-CAM untuk pengambilan gambar saat deteksi terjadi. Proses pemrograman dan komunikasi dengan komputer dilakukan melalui modul FTDI. Mockup ini tidak hanya digunakan untuk pengujian teknis, tetapi juga sebagai alat bantu visual bagi pengguna untuk memahami cara kerja sistem serta memberikan umpan balik sebelum implementasi secara menyeluruh.





Dari komponen dalam system terdapat beberapa rangkaian, seperti:

- A. Sensor MQ-2 berfungsi sebagai input untuk mendeteksi keberadaan gas berbahaya di udara seperti asap rokok. Sensor ini terhubung dengan mikrokontroler ESP32 Devkit V1, yang akan memproses data dan meneruskannya ke sistem monitoring berbasis IoT. Pada



rangkaian ini, pin VCC pada sensor MQ-2 terhubung ke pin 3.3v pada ESP32 Devkit V1 untuk menyuplai tegangan. Pin GND sensor dihubungkan ke GND ESP32 Devkit V1 sebagai jalur *ground*. Sementara itu, pin AO sensor dihubungkan ke pin D34 ESP32 Devkit V1 untuk mengirimkan sinyal analog yang merepresentasikan kadar asap yang terdeteksi.

```
const int MQ2_PIN = 34; // GPIO untuk sensor MQ-2

// Atur resolusi pembacaan analog (dalam setup)
analogReadResolution(12); // Nilai ADC 0-4095

// Fungsi pembacaan dan konversi PPM
float mapAnalogToPPM(int value) {
    float ppm = map(value, 400, 1500, 100, 300); // Mapping analog → PPM

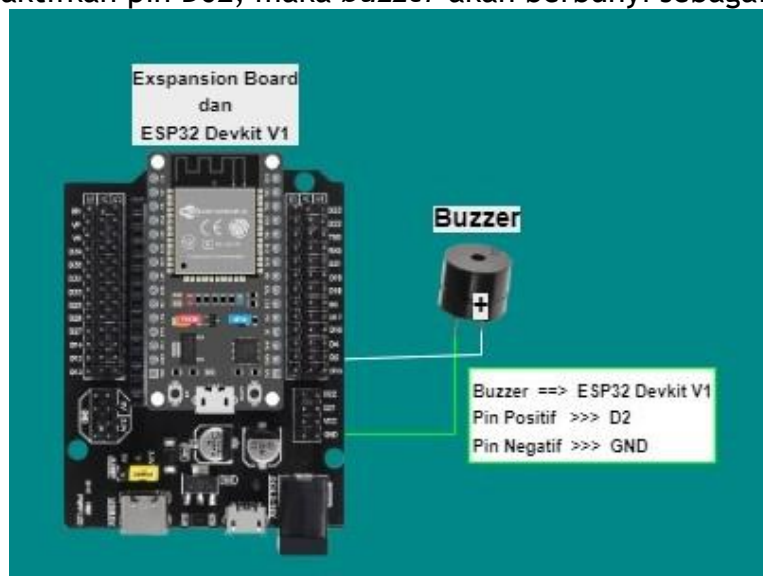
    if (ppm < 0) {
        ppm = 0;
    }

    return ppm;
}

// Pembacaan di loop / fungsi cekAsap()
int analogValue = analogRead(MQ2_PIN); // Baca nilai dari sensor
float ppm = mapAnalogToPPM(analogValue); // Konversi ke estimasi PPM
```


Program MQ-2 mendefinisikan pin GPIO 34 sebagai jalur input dari sensor MQ-2. Pin ini akan digunakan untuk membaca sinyal analog dari pin AO sensor MQ-2, yang mewakili kadar gas (seperti asap rokok) dalam bentuk tegangan. Perintah `analogReadResolution(12);` ini digunakan didalam `setup()` untuk mengatur resolusi pembacaan analog ke 12-bit, artinya pembacaan sensor akan menghasilkan angka antara 0 hingga 4095. Ini memberikan resolusi yang tinggi, yang penting untuk membaca sinyal analog seperti kadar gas secara akurat. Selanjutnya adalah fungsi untuk mengonversi nilai analog yang dibaca dari sensor menjadi estimasi dalam satuan PPM (Parts Per Million). Fungsi `map()` mengubah skala dari nilai analog ke PPM. Dalam hal ini, nilai analog 400 dianggap setara dengan 100 PPM dan nilai analog 1500 dianggap setara dengan 300 PPM. Jika nilai analog kurang dari 400, hasil konversinya bisa negatif, maka diatasi dengan `if (ppm < 0) ppm = 0;` untuk mencegah nilai tidak masuk akal.

- B. Pada rangkaian ini, *buzzer* digunakan sebagai output untuk memberikan peringatan suara saat adanya asap rokok. Pin positif (+) buzzer dihubungkan ke pin D02 pada ESP32 Devkit V1 sebagai pin kontrol digital, sementara pin negatif (-) buzzer dihubungkan ke GND pada ESP32 Devkit V1 sebagai jalur *ground*. Ketika sensor mendeteksi adanya asap rokok dan sistem mengaktifkan pin D02, maka *buzzer* akan berbunyi sebagai tanda *alarm*.



```
const int BUZZER_PIN = 2; // GPIO untuk buzzer

// Inisialisasi di setup
pinMode(BUZZER_PIN, OUTPUT);
digitalWrite(BUZZER_PIN, LOW); // Matikan saat awal

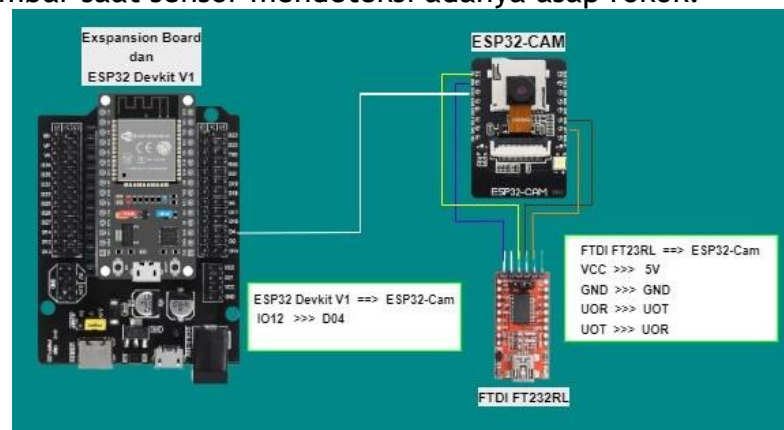
// Variabel pendukung
bool buzzerState = LOW;
unsigned long lastBuzzerToggleTime = 0;
const long BUZZ_ON_DURATION = 200;
const long BUZZ_OFF_DURATION = 300;

// Fungsi non-blocking pengendali buzzer
void kendaliBuzzer() {
    unsigned long currentTime = millis();

    if (isAlarmActive) { // Hanya aktif saat alarm menyala
        if (buzzerState == LOW && currentTime - lastBuzzerToggleTime >= BUZZ_OFF_DURATION) {
            digitalWrite(BUZZER_PIN, HIGH);
            buzzerState = HIGH;
            lastBuzzerToggleTime = currentTime;
        } else if (buzzerState == HIGH && currentTime - lastBuzzerToggleTime >= BUZZ_ON_DURATION) {
            digitalWrite(BUZZER_PIN, LOW);
            buzzerState = LOW;
            lastBuzzerToggleTime = currentTime;
        }
    } else {
        if (buzzerState == HIGH) {
            digitalWrite(BUZZER_PIN, LOW);
            buzzerState = LOW;
        }
    }
}
```

GPIO pin 2 dipakai untuk menghubungkan *buzzer* aktif, dan sistem akan menyalakan dan mematikan *buzzer* dalam pola berkedip jika asap terdeteksi (*isAlarmActive == true*). Baris *pinMode()* mengatur pin sebagai *output*, dan *digitalWrite(LOW)* memastikan *buzzer* tidak aktif saat perangkat baru menyala. Durasi nyala dan mati *buzzer* ditentukan oleh *BUZZ_ON_DURATION* (200 ms) dan *BUZZ_OFF_DURATION* (300 ms). Variabel *buzzerState* digunakan untuk menyimpan status *buzzer* saat ini, sedangkan *lastBuzzerToggleTime* mencatat kapan terakhir *buzzer* berubah status. Fungsi *millis()* dipakai agar proses nyala-mati *buzzer* tidak mengganggu proses penting lain seperti membaca sensor, mengirim pesan Telegram, atau memicu kamera. Ketika alarm asap aktif (*isAlarmActive* bernilai *true*), maka *buzzer* akan berkedip terus-menerus. Ketika tidak ada asap, *buzzer* akan langsung dimatikan sepenuhnya.

- C. Pada rangkaian ini, ESP32 Devkit V1 berfungsi sebagai mikrokontroler utama yang mengatur sistem, membaca data dari sensor, dan mengirim sinyal kontrol ke komponen lain. ESP32 Devkit V1 terhubung ke ESP32-CAM melalui jalur IO12 (ESP32 Devkit V1) yang dihubungkan ke pin D04 (ESP32-CAM) untuk keperluan sinkronisasi atau pemicu pengambilan gambar saat sensor mendeteksi adanya asap rokok.



```
const int CAM_TRIGGER_PIN = 4; // GPIO untuk trigger kamera
const long PHOTO_TRIGGER_HIGH_DURATION = 100; // ms

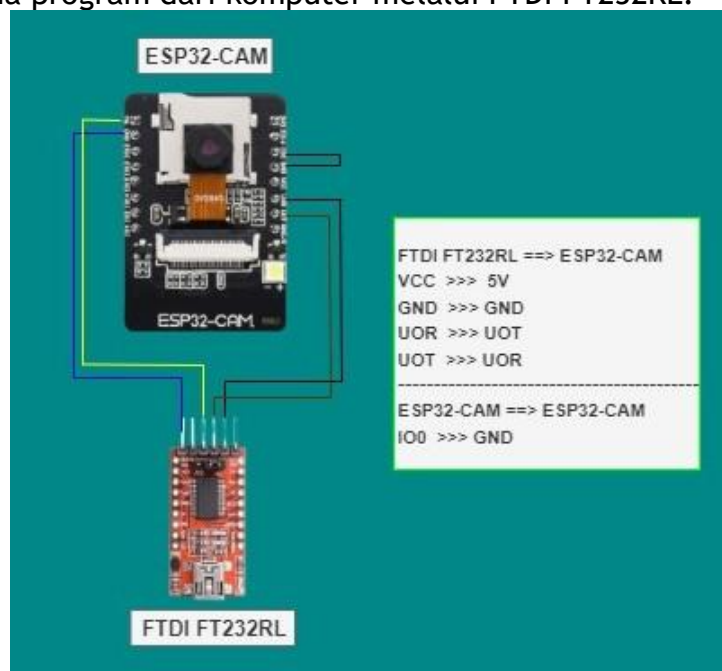
// Inisialisasi di setup
pinMode(CAM_TRIGGER_PIN, OUTPUT);
digitalWrite(CAM_TRIGGER_PIN, LOW); // LOW saat awal

// Fungsi untuk trigger kamera
void triggerKamera() {
  Serial.println("🚬 Memicu ESP32-CAM...");
  digitalWrite(CAM_TRIGGER_PIN, HIGH);
  delay(PHOTO_TRIGGER_HIGH_DURATION); // Kirim sinyal HIGH singkat
  digitalWrite(CAM_TRIGGER_PIN, LOW);
  Serial.println("✅ Trigger selesai.");
}
```

Sementara itu, ESP32-CAM digunakan sebagai modul kamera untuk menangkap gambar atau video ketika sistem mendeteksi keberadaan asap rokok. Agar ESP32-CAM dapat diprogram dan dikonfigurasi, digunakan FTDI FT232RL sebagai USB to Serial adapter. FTDI FT232RL menghubungkan komputer dengan ESP32-CAM, dengan koneksi sebagai berikut, VCC (FTDI) terhubung ke 5V (ESP32-CAM) sebagai sumber daya, GND (FTDI) ke GND (ESP32-CAM) sebagai ground bersama, UOR (FTDI) ke UOT (ESP32-CAM) sebagai jalur komunikasi RX ke TX, dan UOT (FTDI) ke UOR (ESP32-CAM) sebagai jalur TX ke RX.

Const int CAM_TRIGGER_PIN = 4; mendefinisikan bahwa GPIO 4 digunakan untuk mengirim sinyal *trigger* ke ESP32-CAM. *PHOTO_TRIGGER_ HIGH_DURATION = 100;* menentukan berapa lama sinyal *HIGH* dikirim ke pin *trigger* kamera, dalam milidetik. Biasanya cukup antara 50-200ms. *pinMode (CAM_TRIGGER_PIN, OUTPUT);* dan *digitalWrite(CAM_TRIGGER_PIN,LOW);* Mengatur pin sebagai *output* dan memastikan kondisinya *LOW* di awal agar tidak langsung memicu kamera saat *startup*. *Void triggerKamera()* fungsi ini akan dijalankan hanya satu kali ketika asap pertama kali terdeteksi, bukan terus-menerus. Didalam fungsi, *digitalWrite(HIGH)* untuk menyalakan sinyal ke ESP32-CAM, *delay(100)* untuk menjaga sinyal *HIGH* selama 100ms, *digitalWrite(LOW)* untuk mematikan sinyal kembali, dan *Serial.println()* hanya untuk *log* di monitor serial, membantu *debugging*.

- D. Pada rangkaian ini, ESP32-CAM dihubungkan ke FTDI FT232RL yang berfungsi sebagai USB to Serial adapter untuk memprogram ESP32-CAM dari komputer. FTDI FT232RL menghubungkan sinyal USB komputer ke jalur komunikasi serial ESP32-CAM sehingga proses upload program dapat dilakukan. Pin VCC pada FTDI dihubungkan ke 5V pada ESP32-CAM sebagai sumber daya utama, sedangkan pin GND FTDI dihubungkan ke GND ESP32-CAM sebagai jalur ground bersama. Untuk komunikasi data, pin UOR FTDI dihubungkan ke UOT ESP32-CAM (jalur RX ke TX), dan pin UOT FTDI dihubungkan ke UOR ESP32-CAM (jalur TX ke RX). Selain itu, pin IO0 pada ESP32-CAM dihubungkan ke GND agar ESP32-CAM masuk ke mode *flashing* saat dinyalakan. Dengan konfigurasi ini, ESP32-CAM siap menerima program dari komputer melalui FTDI FT232RL.



```

const int CAM_TRIGGER_PIN_IN = 13; // GPIO 13 sebagai pin input trigger dari ESP32 DevKit

pinMode(CAM_TRIGGER_PIN_IN, INPUT_PULLDOWN); // Diset sebagai input dengan resistor pull-down internal

if (digitalRead(CAM_TRIGGER_PIN_IN) == HIGH) {
  Serial.println("Perintah ambil foto diterima dari DevKit!");
  delay(50); // Jeda singkat untuk stabilisasi sinyal

  fb = esp_camera_fb_get(); // Ambil frame dari kamera

  if (!fb) {
    Serial.println("Kamera Gagal Mengambil Gambar!");
    bot.sendMessage(CHAT_ID, "Kamera ESP32-CAM gagal mengambil gambar.", "");
    while (digitalRead(CAM_TRIGGER_PIN_IN) == HIGH) delay(10);
    return;
  }

  dataAvailable = true;
  currentByteIndex = 0;

  Serial.printf("Gambar diambil. Ukuran: %u byte\n", fb->len);
  Serial.println("Mengirim foto ke Telegram...");

  if (bot.sendPhotoByBinary(CHAT_ID, "image/jpeg", fb->len,
    isMoreDataAvailable,
    getNextByte,
    getNextBuffer_for_callback,
    dummyProgressCallback)) {
    Serial.println("Foto berhasil dikirim ke Telegram.");
    bot.sendMessage(CHAT_ID, "📸 Foto Lokasi Asap 📸", "");
  } else {
    Serial.println("Gagal mengirim foto ke Telegram.");
    bot.sendMessage(CHAT_ID, "Gagal mengirim foto ke Telegram.", "");
  }

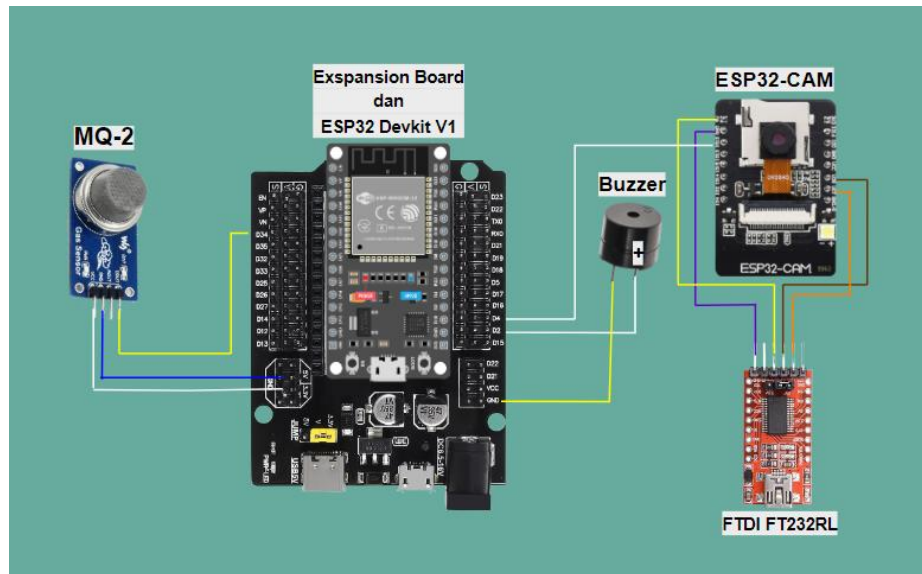
  esp_camera_fb_return(fb); // Kembalikan buffer ke sistem
  fb = NULL;

  while (digitalRead(CAM_TRIGGER_PIN_IN) == HIGH) delay(10); // Tunggu sinyal LOW sebelum lanjut
  Serial.println("Menunggu perintah pemicu berikutnya...");
}

```

Sinyal *trigger* dari ESP32 DevKit melalui pin GPIO 13, disetel sebagai *INPUT_PULLDOWN*, artinya dalam keadaan normal pin akan *LOW*, dan hanya akan terbaca *HIGH* jika diberi sinyal dari luar (dalam hal ini dari DevKit). Ketika ESP32-CAM membaca pin ini sebagai *HIGH*, maka sistem menganggap itu sebagai perintah untuk mengambil gambar. ESP32-CAM menjalankan *esp_camera_fb_get()* untuk mengambil satu frame dari kamera OV2640. Jika proses ini gagal, akan dikirimkan pesan *error* ke Telegram. Jika berhasil, gambar kemudian dikirim melalui fungsi *sendPhotoByBinary()* ke Telegram dengan format JPEG. Fungsi *sendPhotoByBinary()* memerlukan beberapa fungsi *callback* seperti, *isMoreDataAvailable()* untuk mengecek apakah masih ada byte gambar yang perlu dikirim, *getNextByte()* untuk mengirim satu *byte* gambar per panggilan, *getNextBuffer_for_callback()* untuk menyediakan *buffer* lengkap (diminta oleh *library* Telegram meskipun tidak selalu dipakai), dan *dummyProgressCallback()* *Placeholder* progress, tidak aktif tapi diperlukan oleh fungsi. Setelah foto berhasil dikirim, pesan caption seperti "Foto Lokasi Asap" juga dikirimkan ke Telegram untuk memberi konteks gambar. *Frame buffer* dikembalikan ke sistem melalui *esp_camera_fb_return()* agar memori tidak bocor. Sistem lalu menunggu hingga sinyal *trigger* *LOW* kembali, supaya pengambilan gambar tidak terjadi berulang dalam satu sinyal yang sama.

- E. Pada rangkaian ini, ESP32 Devkit V1 berfungsi sebagai pusat pengendali utama sistem deteksi asap rokok. Modul ini terhubung ke beberapa komponen penting, yaitu sensor MQ-2, *buzzer*, ESP32-CAM, dan FTDI FT232RL. MQ-2 digunakan sebagai sensor pendeteksi asap, dengan pin data dihubungkan ke pin input digital pada ESP32 Devkit V1 untuk membaca kadar asap. Ketika ambang batas melebihi yang ditentukan, ESP32 Devkit V1 akan mengaktifkan pin kontrol *buzzer* pada pin D2 untuk memberikan peringatan suara melalui *buzzer*. *Buzzer* dihubungkan dengan pin positif ke D2 dan pin negatif ke GND.



Selain itu, ESP32 Devkit V1 juga terhubung ke ESP32-CAM melalui jalur IO12 (ESP32 Devkit V1) ke D04 (ESP32-CAM) untuk memicu pengambilan gambar saat terjadi deteksi asap. ESP32-CAM terhubung ke FTDI FT232RL yang berfungsi sebagai penghubung ESP32-CAM dengan komputer untuk keperluan pemrograman (*flashing*). Jalur koneksi FTDI ke ESP32-CAM meliputi VCC ke 5V, GND ke GND, UOR ke UOT, dan UOT ke UOR. Sementara itu, IO0 pada ESP32-CAM dihubungkan ke GND agar ESP32-CAM masuk mode *flashing* saat dihidupkan.

Setelah koneksi berhasil dan papan mikrokontroler terdeteksi, unduh serta tambahkan pustaka (*library*) yang dibutuhkan sesuai dengan komponen yang digunakan:

1. ESP32 Devkit V1 dengan MQ-2 dan buzzer:

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h> // Library untuk Telegram

// ==== KONFIGURASI TELEGRAM BOT ====
#define BOT_TOKEN "7690249475:AAG14TrQQ1wNPBJowYTR6H6GKpDBI14RPVI"
#define CHAT_ID "2115182409"

// Kredensial WiFi
const char* ssid = "Sudiono Family";
const char* password = "sdn170167";

WiFiClientSecure client; // Digunakan untuk koneksi HTTPS ke Telegram API
UniversalTelegramBot bot(BOT_TOKEN, client); // Objek bot Telegram
```

- a. <Wi-Fi.h>, untuk koneksi Wi-Fi
- b. <WiFiClientSecure.h>, untuk koneksi aman ke server.
- c. <UniversalTelegramBot.h>, untuk komunikasi dengan bot Telegram agar dapat mengirimkan notifikasi.

2. ESP32-CAM dengan FTDI FT232RL:

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h> // Library ini akan digunakan
#include "esp_camera.h" // Library untuk kamera ESP32
#include <ArduinoJson.h> // Mungkin diperlukan untuk parsing pesan (jika ada fitur bot tambahan)

// ==== KONFIGURASI WIFI (HARUS SAMA DENGAN DEV KIT) ====
#define WIFI_SSID "Sudiono Family" // <-- Ganti dengan nama Wifi Anda (Harus SAMA)
#define WIFI_PASSWORD "sdn170167" // <-- Ganti dengan password Wifi Anda (Harus SAMA)

// ==== KONFIGURASI TELEGRAM BOT (HARUS SAMA DENGAN DEV KIT) ====
#define BOT_TOKEN "7690249475:AAG14TrQQ1wNPBJowYTR6H6GKpDBI14RPVI" // Token Bot dari Botfather
#define CHAT_ID "2115182409" // Chat ID dari @IDBot
```



- a. <WiFi.h>, untuk koneksi *Wi-Fi*.
- b. <WiFiClientSecure.h>, untuk koneksi aman ke server.
- c. "esp_camera.h", untuk mengakses dan mengontrol kamera ESP32-CAM.
- d. <ArduinoJson.h>, untuk memproses data JSON, misalnya saat berkomunikasi dengan API Telegram.

C. Customer Test Drives Mock-Up

Pengujian prototipe terhadap alat guna menilai apakah fungsionalitasnya telah sesuai dengan spesifikasi dan kebutuhan yang diharapkan. Apabila hasil pengujian prototipe menunjukkan bahwa alat belum memenuhi kriteria yang ditetapkan, maka akan dilakukan proses penyempurnaan hingga sistem dapat beroperasi secara optimal sesuai keinginan pengguna. Setelah seluruh tahapan perancangan dan perakitan komponen selesai dilaksanakan, langkah berikutnya adalah melakukan serangkaian pengujian untuk memastikan performa sistem.

Pengujian Blackbox

Metode ini digunakan untuk memastikan bahwa perangkat beroperasi sesuai dengan spesifikasi yang telah ditentukan. Pendekatan ini berfokus pada pemeriksaan apakah semua fitur berfungsi sebagaimana mestinya dari sudut pandang pengguna, tanpa melihat ke dalam struktur atau kode program. Tujuan utama dari metode ini adalah untuk mendeteksi adanya kesalahan atau bug, serta mengonfirmasi bahwa sistem memenuhi persyaratan fungsional dan non-fungsional. Pengujian ini juga dimanfaatkan untuk mengevaluasi kinerja sistem dalam menghadapi beban operasional serta mengukur seberapa cepat sistem merespons perintah. Disamping itu, aspek keamanan juga diuji guna mengidentifikasi potensi risiko, sehingga perangkat lunak dapat berjalan dengan andal sesuai kebutuhan pengguna.

Hasil Uji Coba Alat dan Sistem

No	Nilai Sensor Gas (analog)	Status Sistem	Notifikasi Telegram
1	0-100 PPM	Aman	Tidak Terkirim
2	101-200 PPM	Aman	Tidak Terkirim
3	201-299 PPM	Waspada	Tidak Terkirim
4	300 PPM	Bahaya	Terkirim



Hasil Uji Coba Deteksi Gas

No	Sensor Mendeteksi (nilai analog)	Hasil yang Diharapkan	Hasil Sebenarnya	Kesimpulan
1	0-100 PPM	LED Biru ESP32 Devkit V1 tidak menyala, <i>Buzzer</i> mati, serial monitor menampilkan status " <i>Aman</i> ", notifikasi tidak dikirim.	Sesuai	Valid
2	101-200 PPM	LED Biru ESP32 Devkit V1 tidak menyala, <i>Buzzer</i> mati, serial monitor menampilkan status " <i>Aman</i> ", notifikasi tidak dikirim.	Sesuai	Valid
3	201-299 PPM	LED Biru ESP32 Devkit V1 tidak menyala, <i>Buzzer</i> mati, serial monitor menampilkan status " <i>Waspada</i> ", notifikasi tidak dikirim.	Sesuai	Valid
4	300 PPM	LED Biru ESP32 Devkit V1, <i>Buzzer</i> menyala, serial monitor menampilkan teks <i>scrolling</i> "Asap Terdeteksi! Estimasi PPM: 300 (<i>Buzzer</i> ON)", beserta perintah memicu ESP32-CAM untuk ambil foto dan notifikasi dikirim ke Telegram beserta foto Lokasi yang sudah diambil oleh ESP32-CAM.	Sesuai	Valid

Hasil Uji Coba Indikator Visual dan Audio

No	Kondisi Sensor (nilai analog)	LED ESP32 Devkit V1	ESP32-CAM	Buzzer	Kesimpulan
1	0-100 PPM	LED biru tidak menyala	Tidak aktif	Mati	Valid
2	101-200 PPM	LED biru tidak menyala	Tidak aktif	Mati	Valid
3	201-299 PPM	LED biru tidak menyala	Tidak aktif	Mati	Valid
4	300 PPM	LED biru menyala	Aktif	Menyala	Valid

Hasil Uji Coba Notifikasi Telegram

No	Skenario Pengujian	Output yang Diharapkan	Bukti	Kesimpulan
1	Kadar gas rendah (Aman)	Tidak mengirim notifikasi	-	Valid
2	Kadar gas sedang (Waspada)	Tidak mengirim notifikasi	-	Valid
3	Kadar gas tinggi (Bahaya)	Mengirim notifikasi ke Telegram		Valid

Kesimpulan

Penelitian ini berhasil mengembangkan sebuah sistem deteksi asap rokok dan dokumentasi mahasiswa berbasis *Internet of Things* (IoT) yang terintegrasi dengan aplikasi Telegram dan ESP32-CAM. Sistem ini dirancang untuk digunakan di lingkungan kampus, khususnya area yang rawan pelanggaran.



Beberapa poin kesimpulan yang dapat diambil dari penelitian ini adalah:

1. Sistem Deteksi Asap Rokok Berbasis IoT

Sistem ini mampu mendeteksi keberadaan asap rokok di udara. ESP32 DevKit V1 sebagai mikrokontroler memproses data dari sensor dan mengendalikan keluaran seperti buzzer serta memicu pengambilan gambar menggunakan ESP32-CAM.

2. Fungsi Dokumentasi Mahasiswa secara Otomatis

Ketika terdeteksi asap melebihi ambang batas (300 PPM), sistem secara otomatis mengaktifkan buzzer sebagai peringatan dan memicu ESP32-CAM untuk mengambil foto mahasiswa yang berada di lokasi kejadian. Foto tersebut kemudian dikirimkan melalui aplikasi Telegram ke pihak terkait untuk dokumentasi dan penegakan aturan.

3. Peringatan dan Notifikasi *Real-Time* melalui Telegram

Aplikasi Telegram digunakan sebagai notifikasi berbasis bot secara *real-time*. Pengguna akan menerima informasi berupa pesan teks dan foto kejadian secara otomatis ketika pelanggaran terjadi.

4. Pengujian *Black Box* Berhasil Menunjukkan Kinerja yang Stabil

Berdasarkan metode pengujian *black box*, sistem terbukti mampu mendeteksi asap, mengaktifkan *buzzer*, mengambil gambar, dan mengirimkan notifikasi secara tepat waktu. Sistem bekerja dengan baik dari sisi *input* hingga *output*, sesuai dengan spesifikasi fungsional yang dirancang.

Daftar Pustaka

- Anwari, A., Santoso, L. H., Fitri, R., & ... (2023). Rancang Bangun Alat Pendeteksi Asap Rokok Untuk Penanggulangan Ketertiban Berbasis Internet of Thing. In *INFOTEX: Jurnal Ilmiah* (Vol. 2, Issue 1). <https://ojs.stttexmaco.ac.id/index.php/infotex/article/view/47>
- Aulia, R., Septiana, V., & Mahmudin, W. (2024). *Prototype Alat Monitoring Penyiraman Tanaman Berbasis Iot (Studi Kasus Tanaman Cabai)*. 12(1), 69-77.
- Basuki, A., & Efendi, E. (2022). Prototipe Monitoring dan Penetralisir Asap Rokok Berbasis IoT. *ReTII*, 2022(November), 195-202.
- Governor of Spesial Capital Territory of Jakarta. (2020). Regulation of the Governor of Special Capital Territory of Jakarta Number 40 of 2020 about Amendments to Regulation of the gubernur number 50 of 2012 concerning guidelines for implementing guidance, supervision and law enforcement in no-smoking areas. *Government of of Spesial Capital Territory of Jakarta*.
- Kakihary, N. L. (2021). Pieces Framework for Analysis of User Saticfaction Internet of Things-Based Devices. *Journal of Information Systems and Informatics*, 3(2), 243-252. <https://doi.org/10.33557/journalisi.v3i2.119>
- Muafani, M. (2020). Pemanfaatan Internet of Things (Iot) Pada Desain Rumah Tinggal. *Jurnal Ilmiah Arsitektur*, 10(2), 61-66. <https://doi.org/10.32699/jiars.v10i2.1620>
- Riadi, S., Sunardi, H., Setiawan, C., & Coyanda, J. R. (2024). Pengembangan Prototipe Sistem Monitoring Air Berbasis Internet Of Things Untuk Menghitung Jumlah Konsumsi Dan Biaya Penggunaan. *Journal of Intelligent Networks and IoT Global*, 2(1), 30-38. <https://doi.org/10.36982/jinig.v2i1.4438>
- Sohail Aslam¹, Maqsood Ahmad², H. F. A. and S. E. (2021). Konsep dan Implementasi Internet of Things. In *Journal GEEJ* (Vol. 7, Issue 2). http://www.joi.isoss.net/PDFs/Vol-7-no-2-2021/03_J_ISOSS_7_2.pdf
- Tendra, G., & Wulandari, D. (2020). Alat Pembersih Asap Rokok Otomatis Dengan Menggunakan Sensor Mq2. *Informatika*, 12(1), 1. <https://doi.org/10.36723/juri.v12i1.194>