Integrasi IoT dan Machine Learning pada Edge Device untuk Prediksi Kualitas Udara Perkotaan

Raka Naufal Dzakwan
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
rakanaufal@student.telkomuniversity.ac.id

Hanif Abdillah
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
hanifabdillah@student.telkomuniversity.ac.id

Ahmad Rifada
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
rifada@student.telkomuniversity.ac.id

Ir. Uke Kurniawan Usman, M.T., IPM.
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
ukeusman@telkomuniversity.ac.id

Heru Syah Putra, S.Kom., M.Sc., Eng.
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
herusyahputra@telkomuniversity.ac.id

***Abstrak*— Pencemaran udara merupakan permasalahan lingkungan yang memengaruhi kesehatan masyarakat, terutama di wilayah perkotaan dengan aktivitas kendaraan dan industri yang tinggi. Perubahan kualitas udara yang dinamis meningkatkan kebutuhan akan sistem monitoring real-time yang tidak hanya memantau kondisi udara, tetapi juga mampu melakukan prediksi secara cepat. Penelitian ini mengembangkan sistem monitoring dan prediksi kualitas udara perkotaan yang mengintegrasikan Internet of Things (IoT) dan machine learning pada edge device. Sistem menggunakan ESP32 untuk akuisisi data lingkungan, Firebase Realtime Database sebagai media penyimpanan data, serta Raspberry Pi 4 sebagai edge device yang menjalankan model Random Forest Regressor. Parameter yang digunakan meliputi PM1.0, PM2.5, PM10, MQ-135, MQ-7, suhu, dan kelembapan yang dikumpulkan pada tiga lokasi dengan karakteristik lingkungan berbeda. Data hasil pengukuran digunakan untuk memprediksi konsentrasi PM2.5 dan ditampilkan melalui aplikasi AtmoSense berbasis FlutterFlow. Hasil pengujian menunjukkan bahwa sistem mampu melakukan akuisisi, pengiriman, penyimpanan, visualisasi, dan prediksi data secara terintegrasi. Model Random Forest Regressor menghasilkan R^2 sebesar 99,73%, MAE sebesar 1,5464 $\mu g/m^3$, MSE sebesar 9,4977 $\mu g^2/m^6$, dan RMSE sebesar 3,0818 $\mu g/m^3$. Pengujian QoS menghasilkan throughput 45,15 kbps, delay 54,84 ms, jitter 0,33 ms, dan packet loss 0,1%, yang termasuk kategori sangat baik berdasarkan TIPHON. Konsumsi daya keseluruhan sistem terukur sebesar 17,80 W.*

***Kata kunci*—edge computing, Internet of Things, kualitas udara, PM2.5, Random Forest**

I. PENDAHULUAN

Kualitas udara merupakan salah satu indikator utama yang mencerminkan kondisi lingkungan dan tingkat kesehatan masyarakat. Pertumbuhan kawasan perkotaan, aktivitas transportasi, pembangunan industri, dan kepadatan penduduk meningkatkan konsentrasi polutan di udara, termasuk partikulat (PM1.0, PM2.5, PM10) dan gas pencemar seperti karbon monoksida (CO), amonia (NH_3), benzena, dan asap hasil pembakaran [1]. Paparan polutan secara terus-menerus dapat menyebabkan iritasi saluran pernapasan, gangguan fungsi paru, serta penyakit kronis seperti asma dan kardiovaskular; partikulat halus seperti PM2.5 bahkan dapat masuk ke aliran darah sehingga meningkatkan risiko penyakit serius.

Perkembangan teknologi Internet of Things (IoT) memungkinkan akuisisi data lingkungan dilakukan secara otomatis dan real-time melalui integrasi sensor, mikrokontroler, dan konektivitas internet. Data yang diperoleh dapat dikirim ke cloud untuk disimpan, diolah, dan divisualisasikan melalui aplikasi monitoring, sehingga kondisi lingkungan dapat diakses dari berbagai lokasi. Namun, sebagian besar sistem yang ada masih berfokus pada pemantauan kondisi saat ini dan belum mampu memberikan informasi prediksi untuk kondisi mendatang [2]. Kemampuan prediksi penting untuk mendukung upaya mitigasi, memungkinkan masyarakat mengambil tindakan preventif, serta membantu instansi pemerintah merumuskan kebijakan pengendalian kualitas udara yang lebih efektif.

Di sisi lain, meningkatnya jumlah sensor dan volume data yang dihasilkan secara berkelanjutan menimbulkan tantangan dalam pengolahan data. Sistem yang sepenuhnya bergantung pada cloud menghadapi keterbatasan seperti latensi komunikasi, kebutuhan bandwidth yang tinggi, dan ketergantungan pada koneksi internet yang stabil [2]. Kondisi tersebut dapat menghambat analisis data secara real-time, terutama pada aplikasi yang membutuhkan respons cepat. Oleh karena itu, diperlukan pendekatan komputasi yang memproses data lebih dekat dengan sumber data agar efisiensi dan responsivitas sistem meningkat.

Berdasarkan permasalahan tersebut, penelitian ini mengusulkan pengembangan sistem monitoring dan prediksi kualitas udara perkotaan yang mengintegrasikan IoT, edge computing, dan machine learning. Sistem dirancang menggunakan sensor PMS7003 untuk mengukur konsentrasi PM1.0, PM2.5, dan PM10, sensor MQ-7 untuk mendeteksi karbon monoksida (CO), sensor MQ-135 untuk memantau kualitas udara berdasarkan berbagai gas pencemar, serta sensor DHT22 untuk mengukur suhu dan kelembapan lingkungan [3]. Data yang diperoleh dikirim menggunakan mikrokontroler ESP32 menuju Raspberry Pi sebagai edge device yang menjalankan model machine learning berbasis Random Forest [3]. Melalui integrasi ini, sistem diharapkan mampu melakukan monitoring dan prediksi kualitas udara secara real-time dengan respons yang lebih cepat, efisien, dan andal dibandingkan pendekatan monitoring konvensional.

Kontribusi utama penelitian ini adalah sebagai berikut.

1. Mengembangkan sistem monitoring kualitas udara berbasis IoT yang mengintegrasikan ESP32, Raspberry

Pi 4 sebagai *edge device*, Firebase Realtime Database, dan aplikasi AtmoSense.
2. Mengimplementasikan Random Forest Regressor untuk memprediksi konsentrasi PM2.5 berdasarkan data sensor lingkungan.
3. Menerapkan inferensi *machine learning* pada Raspberry Pi 4 untuk mendukung pemrosesan data secara lokal dengan latensi rendah.
4. Mengevaluasi sistem melalui pengujian hardware, software, performa *machine learning*, Quality of Service (QoS), dan konsumsi daya.

II. KAJIAN TEORI

A. Kualitas Udara

Kualitas udara menggambarkan kondisi udara pada suatu wilayah berdasarkan konsentrasi zat dan partikel yang terdapat di atmosfer. Kualitas udara yang baik ditandai dengan rendahnya konsentrasi polutan sehingga tidak menimbulkan dampak negatif terhadap kesehatan manusia maupun lingkungan [4]. Kualitas udara dipengaruhi oleh faktor alami, seperti cuaca, angin, suhu, kelembapan, kebakaran hutan, dan aktivitas vulkanik, maupun faktor antropogenik seperti transportasi, aktivitas industri, pembakaran bahan bakar fosil, dan aktivitas rumah tangga [5]. Pemantauan kualitas udara secara berkelanjutan diperlukan untuk mengetahui kondisi lingkungan secara aktual serta mendukung pengambilan keputusan terkait kesehatan masyarakat dan pengelolaan lingkungan.

B. Internet of Things (IoT)

Internet of Things (IoT) adalah konsep yang menghubungkan perangkat fisik melalui jaringan komunikasi sehingga dapat bertukar data dan bekerja secara terintegrasi tanpa memerlukan intervensi manusia secara terus-menerus [6][7]. Sistem IoT umumnya terdiri atas sensor, mikrokontroler atau perangkat komputasi, jaringan komunikasi, serta platform penyimpanan dan pengolahan data. IoT semakin banyak dikombinasikan dengan machine learning sehingga sistem tidak hanya mampu memantau kondisi lingkungan secara real-time, tetapi juga menganalisis pola data historis dan menghasilkan informasi prediktif untuk kondisi mendatang.

C. Machine Learning

Machine learning merupakan cabang Artificial Intelligence (AI) yang memungkinkan sistem komputer mempelajari pola dari data dan menghasilkan keputusan atau prediksi tanpa perlu diprogram secara eksplisit untuk setiap kasus [8]. Melalui pelatihan pada data historis, model machine learning dapat mengidentifikasi hubungan antarvariabel dan memanfaatkan pola tersebut untuk memprediksi data baru. Pada sistem monitoring kualitas udara, machine learning menganalisis data dari sensor lingkungan untuk mengidentifikasi pola perubahan kualitas udara sehingga informasi yang dihasilkan mendukung pengambilan keputusan yang lebih proaktif, bukan sekadar monitoring [9].

1) Edge Device: Edge device adalah perangkat komputasi yang melakukan pemrosesan data dekat dengan sumber data sebelum data dikirim ke server atau cloud [10]. Konsep *edge computing* memindahkan pemrosesan data lebih dekat ke sumber data untuk mengurangi latensi dan meningkatkan kecepatan respons. Pada penelitian ini, Raspberry Pi 4 berfungsi sebagai *edge device* yang menjalankan model Random Forest Regressor untuk memprediksi konsentrasi PM2.5 secara *real-time*.

2) Random Forest: Random Forest merupakan algoritma ensemble learning yang menggabungkan beberapa model untuk menghasilkan prediksi yang lebih baik dibandingkan model tunggal [11]. Algoritma ini dibangun dari sejumlah decision tree yang masing-masing dilatih pada subset data berbeda melalui bootstrap sampling, dengan keluaran seluruh pohon digabungkan melalui rata-rata (averaging) untuk menghasilkan output regresi akhir. Pendekatan ini meningkatkan stabilitas prediksi, mengurangi risiko overfitting, dan menjaga performa yang robust ketika memproses data lingkungan yang kompleks dan mengandung noise. Random Forest karena itu sesuai untuk mengestimasi konsentrasi PM2.5 berdasarkan parameter partikulat, sensor gas, dan meteorologi, serta menawarkan waktu komputasi yang praktis untuk diterapkan pada edge device yang harus beroperasi secara real-time.

D. Sistem Monitoring Kualitas Udara Berbasis IoT

Sistem monitoring kualitas udara dirancang untuk mengamati, mengukur, dan mengumpulkan informasi kondisi udara suatu lingkungan secara berkelanjutan [12]. IoT telah mentransformasi sistem tersebut dengan memungkinkan data pengukuran dikirim secara otomatis melalui jaringan komunikasi menuju sistem penyimpanan atau platform monitoring, sehingga memungkinkan pemantauan real-time dan akses informasi yang lebih mudah dari berbagai lokasi [13]. Beberapa platform yang telah ada, seperti Biru Langit (dikembangkan oleh Universitas Telkom) dan Nafas, telah menyediakan dashboard monitoring kualitas udara secara real-time. namun, kemampuan prediksi di tingkat edge device masih terbatas pada sebagian besar sistem tersebut, dengan analisis lanjutan yang masih bergantung pada infrastruktur server atau cloud.

E. Sensor Lingkungan

Sensor lingkungan mendeteksi dan mengukur parameter fisik atau kimia di lingkungan sekitar, kemudian mengubah hasil pengukuran menjadi data yang dapat diproses oleh sistem elektronik [13]. Sensor partikulat mengukur konsentrasi partikel padat atau cair yang melayang di udara, umumnya diklasifikasikan berdasarkan diameter (PM1.0, PM2.5, PM10) [5]. Sensor gas mendeteksi keberadaan dan perubahan konsentrasi gas tertentu dengan mengubah perubahan fisik atau kimia akibat paparan gas menjadi sinyal listrik [13]. Sensor suhu dan kelembapan mengukur kondisi termal dan kandungan uap air di udara, yang memengaruhi kenyamanan manusia sekaligus dispersi polutan di atmosfer [12].

F. Penyimpanan Berbasis Cloud

Penyimpanan berbasis cloud memungkinkan data disimpan pada server yang terhubung internet sehingga dapat diakses dari berbagai lokasi dan perangkat [14]. Berbeda dengan penyimpanan lokal yang hanya dapat diakses dari perangkat tertentu, cloud storage menawarkan fleksibilitas dalam pengelolaan data, skalabilitas untuk volume data besar, dan kemudahan sinkronisasi antarperangkat terhubung, yang

menjadikannya banyak digunakan pada sistem IoT dan monitoring lingkungan modern.

G. Mikrokontroler

Mikrokontroler adalah sistem komputasi berukuran kecil yang terdiri atas prosesor, memori, dan antarmuka input/output yang terintegrasi dalam satu chip [15]. Perangkat ini menjalankan tugas-tugas yang telah diprogram secara otomatis dan berfungsi sebagai pengendali utama yang menerima data sensor, melakukan pemrosesan dasar, serta meneruskan data atau perintah ke perangkat lain sesuai kebutuhan. Konsumsi daya yang rendah, ukuran yang kompak, serta kemudahan integrasi dengan sensor menjadikan mikrokontroler komponen inti pada sistem IoT dan otomasi.

III. METODE

A. Spesifikasi Sistem dan Dasar Regulasi

Spesifikasi sistem dirumuskan berdasarkan regulasi nasional yang mengatur penilaian kualitas udara di Indonesia, yaitu Peraturan Menteri Lingkungan Hidup dan Kehutanan Republik Indonesia Nomor P.14/MENLHK/SETJEN/KUM.1/7/2020 tentang Indeks Standar Pencemar Udara (ISPU) [16]. Berdasarkan regulasi tersebut, parameter ISPU meliputi PM10, PM2.5, CO, NO_2, SO_2, O_3, dan hidrokarbon (HC). Penelitian ini berfokus pada PM2.5 sebagai parameter kritis yang dapat diukur dengan sensor berbiaya rendah, dilengkapi suhu dan kelembapan sebagai fitur meteorologi.

Nilai ISPU diperoleh melalui interpolasi linier antara konsentrasi polutan terukur dan rentang indeks yang bersesuaian, sebagaimana Persamaan (1) [16]:

$$I = [(Ia - Ib)/(Xa - Xb)] \times (Xx - Xb) + Ib$$

dengan I adalah ISPU terhitung, Ia dan Ib adalah batas indeks atas dan bawah, Xa dan Xb adalah konsentrasi ambien batas atas dan bawah yang bersesuaian, serta Xx adalah konsentrasi ambien terukur (µg/m³).

TABEL I. KATEGORI ISPU

Kategori	Nilai Indeks	Warna
Baik	0 – 50	Hijau
Sedang	51 – 100	Biru
Tidak Sehat	101 – 200	Kuning
Sangat Tidak Sehat	201 – 300	Merah
Berbahaya	>300	Hitam

TABEL II. AMBANG KONSENTRASI PM2.5 (PERMEN LHK NO. P.14/2020)

Kategori ISPU	Konsentrasi PM2.5 (µg/m³)
Baik	0 – 15,5
Sedang	15,6 – 55,4
Tidak Sehat	55,5 – 150,4
Sangat Tidak Sehat	150,5 – 250,4
Berbahaya	>250,4

B. Spesifikasi Perangkat Keras dan Perangkat Lunak

TABEL III. SPESIFIKASI PERANGKAT KERAS

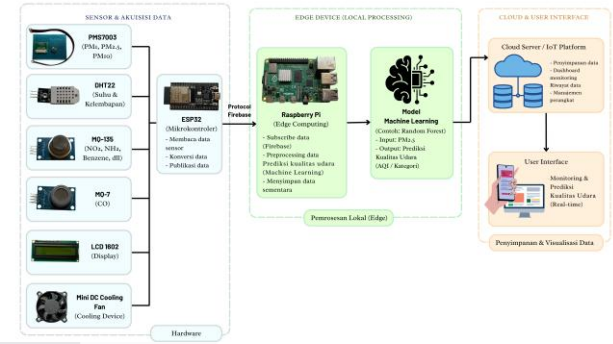
Komponen	Spesifikasi	Fungsi
Sensor partikulat	Pengukuran PM1.0, PM2.5, PM10	Monitoring partikulat
Sensor gas	Monitoring konsentrasi gas pencemar	Pengukuran kualitas udara
Sensor suhu/kelembapan	Suhu dan kelembapan ambien	Parameter meteorologi
Router/Wi-Fi	IEEE 802.11	Komunikasi data
Mikrokontroler	Mendukung Wi-Fi	Akuisisi dan pengiriman data
Edge device	Single-board computer	Pemrosesan data dan prediksi

TABEL IV. SPESIFIKASI PERANGKAT LUNAK

Software	Fungsi
Arduino IDE	Pemrograman mikrokontroler
Python	Pengolahan data dan machine learning
Firebase	Penyimpanan dan sinkronisasi data
FlutterFlow	Pengembangan aplikasi monitoring

Cakupan perangkat keras dibatasi pada satu mikrokontroler yang terhubung ke seluruh sensor, dengan inferensi machine learning dijalankan sepenuhnya pada edge device tanpa server komputasi khusus, dan komunikasi data melalui jaringan Wi-Fi yang terhubung internet. Sistem diimplementasikan pada skala prototipe untuk evaluasi pada lingkungan pengujian terbatas.

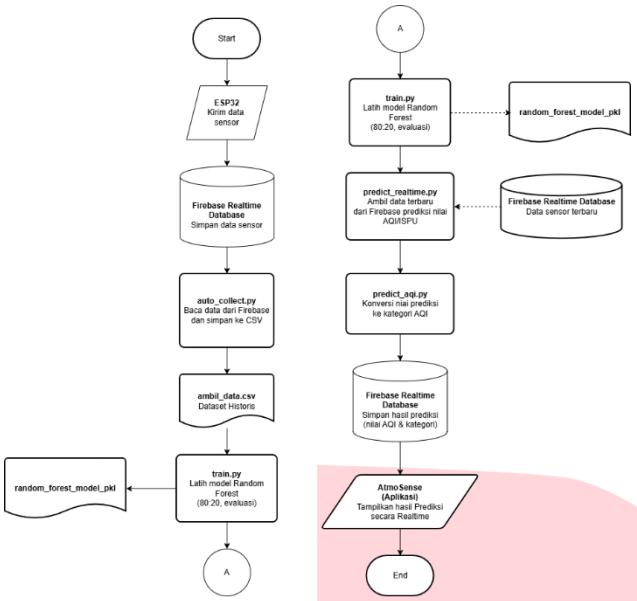
C. Arsitektur Sistem



Gambar 1. Arsitektur keseluruhan sistem monitoring dan prediksi kualitas udara berbasis IoT.

ESP32 mengakuisisi data lingkungan dari sensor PMS7003, MQ-7, MQ-135, dan DHT22, kemudian mengirimkan hasil pembacaan melalui Wi-Fi menuju Firebase Realtime Database. Raspberry Pi 4 mengambil data yang tersimpan dan menjalankan model Random Forest Regressor secara lokal untuk memprediksi konsentrasi PM2.5. Hasil prediksi selanjutnya dikirim kembali ke Firebase dan ditampilkan melalui aplikasi mobile AtmoSense. Arsitektur ini memungkinkan akuisisi data, inferensi machine learning berbasis edge, sinkronisasi cloud, dan visualisasi mobile beroperasi secara berkelanjutan dalam satu sistem terintegrasi.

D. Alur Kerja Monitoring dan Machine Learning



Gambar 2. Alur kerja monitoring dan machine learning pada sistem yang diusulkan.

Alur aplikasi AtmoSense diawali dengan inisialisasi dan pengambilan data dari Firebase Realtime Database, dilanjutkan dengan pembaruan real-time secara berkelanjutan sehingga halaman dashboard, location, prediction, dan device information selalu mencerminkan nilai sensor dan prediksi terbaru.

Alur machine learning diawali dengan ESP32 mengirim data sensor ke Firebase. Skrip pengumpulan otomatis (auto_collect.py) mengagregasi data yang masuk menjadi dataset historis, yang digunakan untuk melatih model Random Forest (train_model.py) menggunakan pembagian data latih-uji 80:20; model yang dihasilkan diserialisasi. Pada saat operasi, skrip prediksi real-time (predict_realtime.py) mengambil data sensor terbaru dari Firebase dan memasukkannya ke model terlatih, setelah itu skrip konversi (predict_ispu.py) mengubah nilai prediksi PM2.5 menjadi kategori ISPU. Hasil prediksi dan kategori dikirim kembali ke Firebase dan selanjutnya ditampilkan oleh AtmoSense secara real-time.

E. Parameter Pengukuran

TABEL V. FUNGSI PARAMETER PENGUKURAN

Parameter	Peran dalam Sistem
PM1.0	Fitur masukan Random Forest Regressor
PM2.5	Target prediksi dan parameter utama evaluasi kualitas udara
PM10	Fitur masukan dan parameter pendukung evaluasi
MQ-7 (CO)	Fitur masukan representasi konsentrasi CO
MQ-135	Fitur masukan representasi konsentrasi gas pencemar
Suhu	Fitur masukan meteorologi
Kelembapan	Fitur masukan meteorologi

PM2.5 berperan sebagai parameter utama, yaitu target evaluasi ISPU sekaligus target prediksi model Random Forest Regressor, sedangkan PM1.0, PM10, MQ-7 (CO), MQ-135 (multi-gas), suhu, dan kelembapan berperan sebagai fitur masukan pendukung yang membantu model mempelajari hubungan antara konsentrasi partikulat, gas pencemar, dan kondisi meteorologi.

F. Metode Verifikasi Perangkat Keras dan Perangkat Lunak

Verifikasi perangkat keras mencakup unit pemrosesan utama, modul akuisisi partikulat, gas, dan meteorologi, serta konsumsi daya sistem, dengan setiap komponen diamati stabilitas dan konsistensi pembacaannya pada berbagai kondisi lingkungan. Verifikasi perangkat lunak mencakup tiga aspek: (i) pengujian Quality of Service (QoS) pada tautan komunikasi antara perangkat IoT dan Firebase Realtime Database, dievaluasi melalui throughput, delay, jitter, dan packet loss; (ii) pengujian fungsional black-box pada seluruh fitur platform AtmoSense; dan (iii) evaluasi performa model Random Forest menggunakan pembagian dataset, R^2, MAE, MSE, RMSE, dan waktu inferensi.

1) Konversi Sinyal Sensor Gas: Keluaran analog sensor MQ-7 dan MQ-135 dibaca ESP32 melalui ADC 12-bit dan dikonversi menjadi tegangan keluaran menggunakan Persamaan (2):

$$Vout = (ADC \times VCC)/4095$$

dengan VCC = 3,33 V sebagai tegangan referensi dan 4095 sebagai resolusi maksimum ADC 12-bit. Resistansi sensor Rs kemudian diperoleh dari Persamaan (3), menggunakan resistor beban RL = 3300 Ω:

$$Rs = [(VCC - Vout)/Vout] \times RL$$

Rasio Rs terhadap resistansi referensi Ro dihitung sebagai Rs/Ro dan disubstitusikan ke kurva kalibrasi spesifik sensor untuk memperoleh konsentrasi gas dalam ppm. Untuk sensor MQ-7 (Ro = 60 Ω), konsentrasi CO diestimasi menggunakan Persamaan (4):

$$COppm = 99{,}042 \times (Rs/Ro)^{-}1{,}518$$

dan untuk sensor MQ-135, konsentrasi multi-gas diestimasi menggunakan Persamaan (5):

$$MQ135ppm = 116{,}6020682 \times (Rs/Ro)^{-}2{,}76903487$$

2) Konsumsi Daya: Konsumsi daya sistem diperoleh dengan mengukur tegangan kerja V dan arus I sistem pada kondisi operasi normal, kemudian menghitung daya listrik menggunakan Persamaan (6):

$$P = V \times I$$

3) Metrik Evaluasi Model: Model Random Forest Regressor dievaluasi menggunakan koefisien determinasi (R^2), Mean Absolute Error (MAE), Mean Squared Error (MSE), dan Root Mean Squared Error (RMSE), berturut-turut pada Persamaan (7)–(10):

$$R^2 = 1 - [\Sigma(yi - \hat{y}i)^2/\Sigma(yi - \bar{y})^2]$$

$$MAE = 1/n\Sigma yi - \hat{y}i$$

$$MSE = (1/n)\Sigma(yi - \hat{y}i)^2$$

$$RMSE = \sqrt{[(1/n)\Sigma(yi - \hat{y}i)^2]}$$

dengan yi adalah nilai aktual, ŷi adalah nilai prediksi, dan ȳ adalah rata-rata nilai aktual. Waktu inferensi diukur sebagai selisih waktu antara data masukan diterima model dan hasil prediksi dihasilkan, menunjukkan kesesuaian model untuk operasi real-time pada edge device.

G. Implementasi Perangkat Keras



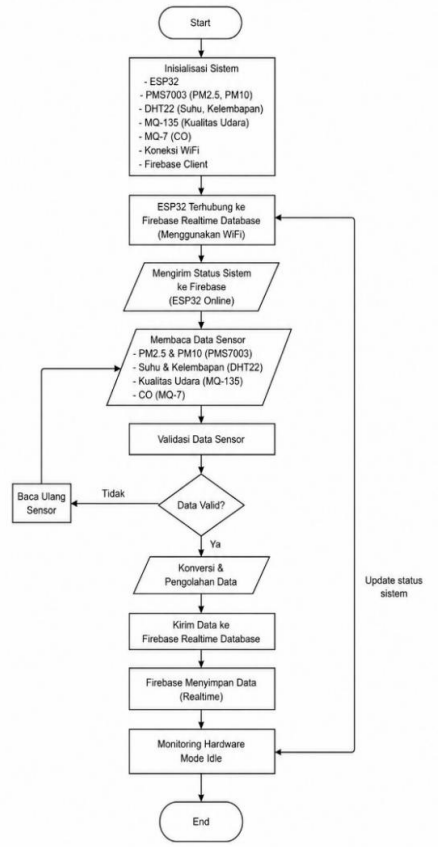
Gambar 3(a). Implementasi fisik perangkat monitoring — tampak depan.



Gambar 3(b). Implementasi fisik perangkat monitoring — tampak dalam.



Gambar 4. Diagram blok perangkat keras sistem.

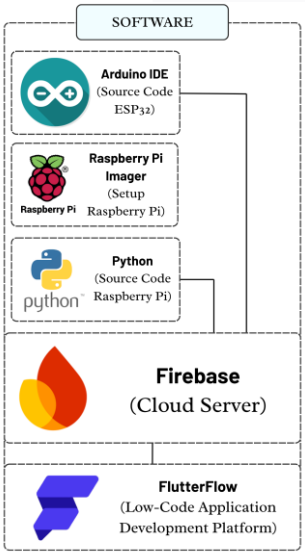


Gambar 5. Alur kerja perangkat keras sistem.

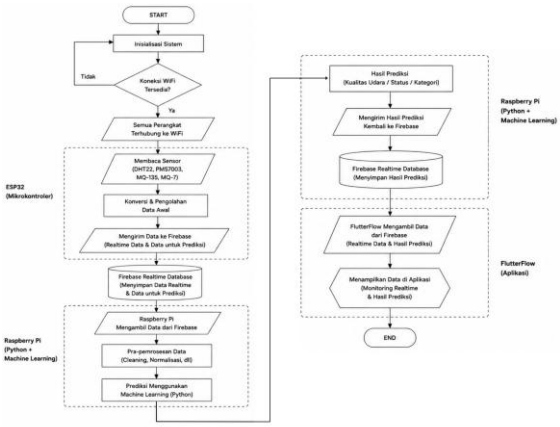
Alur kerja perangkat keras dimulai dengan inisialisasi ESP32, sensor, LCD, kipas pendingin, dan Raspberry Pi. ESP32 kemudian terhubung ke jaringan Wi-Fi untuk melakukan akuisisi data dari PMS7003, MQ-7, MQ-135, dan DHT22. Data sensor diproses dan dikirim ke Firebase Realtime Database, kemudian diambil oleh Raspberry Pi untuk *preprocessing* dan inferensi menggunakan model *machine learning*. Hasil prediksi selanjutnya dikonversi ke kategori ISPU dan ditampilkan melalui *dashboard monitoring* secara *real-time*.

ESP32 merupakan mikrokontroler System-on-Chip (SoC) yang dilengkapi konektivitas Wi-Fi dan Bluetooth [19], banyak digunakan pada sistem IoT karena kemampuan pemrosesan yang memadai, konsumsi daya rendah, serta kemudahan integrasi dengan sensor dan periferal. Sensor PMS7003 berbasis laser-scattering memberikan pembacaan PM1.0, PM2.5, dan PM10 [20][21]. Sensor MQ-7 berbasis semikonduktor mendeteksi karbon monoksida melalui perubahan resistansi yang proporsional terhadap paparan CO [22][23][24]. Sensor MQ-135 mendeteksi rentang gas pencemar yang lebih luas, termasuk amonia, benzena, dan CO_2 [25][24]. Sensor DHT22 memberikan pembacaan suhu dan kelembapan relatif tanpa memerlukan konversi analog-ke-digital [26][27]. Raspberry Pi 4 berfungsi sebagai unit edge computing yang menjalankan Random Forest Regressor terlatih secara lokal, menghindari overhead latensi dan bandwidth pengiriman data mentah sensor ke server jarak jauh untuk inferensi [28][29]. Regulasi daya disediakan oleh modul step-down DC-DC LM2596 yang disuplai adaptor 12 V/3 A [30], didistribusikan melalui PCB dot-matrix custom dua sisi [31][32], dengan data dan daya ke Raspberry Pi dihubungkan melalui kabel USB-to-micro-USB [33].

H. Implementasi Perangkat Lunak

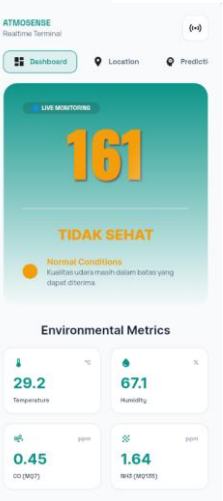


Gambar 6. Arsitektur perangkat lunak sistem.

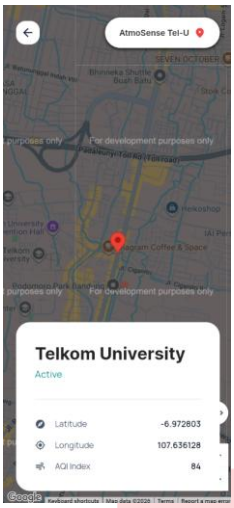


Gambar 7. Alur kerja perangkat lunak sistem.

Firmware ESP32 dikembangkan pada Arduino IDE [34][35] dan secara berkala mempublikasikan pembacaan sensor ke Firebase Realtime Database [36][37]. Pada sisi edge device, skrip Python menangani pipeline machine learning [38]: auto_collect.py secara berkelanjutan mengagregasi data Firebase menjadi dataset historis; train_model.py melatih Random Forest Regressor secara offline menggunakan pembagian data latih-uji 80:20 dan menyerialisasi model; predict_realtime.py memuat model terlatih dan melakukan inferensi pada data sensor terbaru; predict_ispu.py mengonversi nilai prediksi PM2.5 menjadi kategori ISPU sebelum menuliskan hasil kembali ke Firebase.



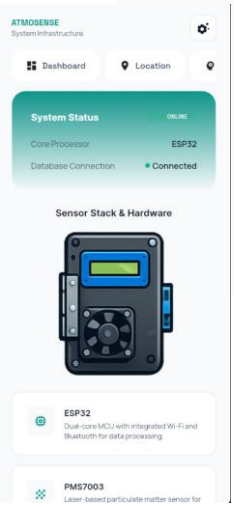
Gambar 8(a). Halaman Dashboard aplikasi AtmoSense.



Gambar 8(b). Halaman Location aplikasi AtmoSense.



Gambar 8(c). Halaman Prediction aplikasi AtmoSense.



Gambar 8(d). Halaman Device Information aplikasi AtmoSense.

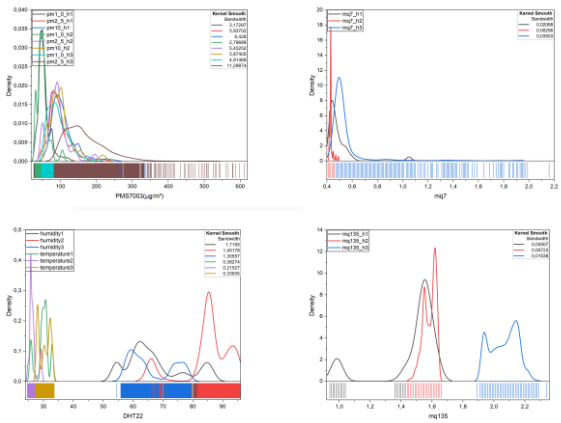
Aplikasi monitoring AtmoSense dikembangkan menggunakan FlutterFlow, platform pengembangan aplikasi low-code berbasis Flutter dengan desain UI visual dan integrasi native ke Firebase [39]. Aplikasi terdiri atas empat halaman utama: Dashboard menampilkan nilai ISPU real-time dan kategori kualitas udara dengan indikator warna, beserta suhu, kelembapan, konsentrasi CO (MQ-7), konsentrasi multi-gas (MQ-135), dan pembacaan PM1.0/PM2.5/PM10 (PMS7003), serta status konektivitas node sensor; Location menampilkan posisi geografis perangkat pada peta digital, beserta nama lokasi, status perangkat, latitude/longitude, dan nilai ISPU pada lokasi tersebut; Prediction menampilkan nilai ISPU dan kategori kualitas udara hasil prediksi model Random Forest, beserta tingkat kepercayaan model dan tabel referensi kategori ISPU; Device Information menampilkan status operasional sistem, status konektivitas database Firebase, dan komponen perangkat keras yang digunakan (ESP32-WROOM-32, PMS7003, MQ-7, MQ-135, dan DHT22). Seluruh halaman diperbarui otomatis seiring tersedianya data baru pada Firebase Realtime Database, memungkinkan monitoring dan akses prediksi secara real-time dari perangkat mobile.

IV. HASIL DAN PEMBAHASAN

A. Skema Pengujian

Pengambilan data dilakukan selama tiga hari pada tiga lokasi dengan karakteristik berbeda, yaitu Jalan Telekomunikasi Telkom University, Green House Telkom University, dan kawasan Braga Kota Bandung, dengan durasi lima jam pada setiap lokasi. Sistem mengukur parameter kualitas udara, suhu, dan kelembapan secara *real-time*. Data dikirim oleh ESP32 ke Firebase Realtime Database untuk ditampilkan melalui AtmoSense dan digunakan oleh Raspberry Pi 4 untuk melakukan prediksi berbasis Random Forest Regressor yang selanjutnya dibandingkan dengan data aktual.

B. Karakteristik Data Sensor



Gambar 9. KDE data seluruh sensor dari seluruh lokasi pengujian.

Karakteristik data sensor dianalisis menggunakan *Kernel Density Estimation* (KDE) untuk menggambarkan distribusi data selama proses pengukuran. PMS7003 menunjukkan variasi partikulat yang lebih luas antarlokasi, sedangkan MQ-7 dan MQ-135 memiliki distribusi yang relatif lebih sempit. Suhu dan kelembapan menunjukkan variasi moderat sesuai perbedaan kondisi lingkungan. Secara keseluruhan, dataset mencakup kondisi kualitas udara yang beragam dan digunakan sebagai data untuk pelatihan serta evaluasi Random Forest Regressor.

C. Analisis Konsumsi Daya

TABEL VI. TEGANGAN, ARUS, DAN DAYA KOMPONEN SISTEM

Komponen	Tegangan (V)	Arus (A)	Daya (W)
ESP32	3,3	0,24	0,79
PMS7003	5,0	0,10	0,50

MQ-135	5,0	0,15	0,75
MQ-7	5,0	0,15	0,75
DHT22	3,3	0,0025	0,008
Raspberry Pi 4	5,5	3,00	15,00

Gabungan daya node sensor (Psensor = 0,79 + 0,50 + 0,75 + 0,75 + 0,008 = 2,798 W) ditambah Raspberry Pi 4 (PRPi = 15,00 W) menghasilkan total konsumsi daya sistem sebesar Ptot = 17,798 W. Raspberry Pi 4 menyumbang sebagian besar anggaran daya karena menjalankan sistem operasi Linux, menampung model Random Forest, serta melakukan inferensi machine learning secara lokal, sementara ESP32 dan rangkaian sensor bersama-sama hanya menyumbang sebagian kecil dari total kebutuhan daya. Kebutuhan daya ini tergolong moderat untuk sistem monitoring IoT dan edge computing serta menyisakan margin yang cukup untuk penambahan komponen di masa mendatang.

D. Analisis *Quality of Service (QoS)*

Lalu lintas jaringan antara ESP32 dan Firebase Realtime Database ditangkap menggunakan Wireshark dan dianalisis terhadap standar TIPHON.

TABEL VII. HASIL PENGUKURAN PARAMETER QOS

Parameter	Nilai
Throughput	45,15 kbps
Delay	54,84 ms
Jitter	0,33 ms
Packet Loss	0,1%

Karena data yang dikirim hanya berupa nilai parameter sensor, *throughput* yang diperoleh mencukupi kebutuhan sistem. Hasil pengujian menunjukkan *delay* sebesar 54,84 ms, *jitter* 0,33 ms, dan *packet loss* 0,1%. Berdasarkan standar TIPHON, seluruh parameter QoS termasuk kategori sangat baik, sehingga komunikasi antara perangkat IoT dan Firebase Realtime Database mampu mendukung monitoring kualitas udara secara *real-time*.

E. Pengujian Fungsional AtmoSense

Pengujian black-box terhadap aplikasi AtmoSense mengonfirmasi bahwa halaman Dashboard, Location, Prediction, dan Device Information seluruhnya beroperasi sesuai fungsi yang dirancang, dengan pengambilan dan penampilan data sensor serta prediksi secara real-time dari Firebase Realtime Database, pembaruan otomatis, dan navigasi antarhalaman yang berjalan dengan benar.

F. Performa Model Machine Learning

Dataset gabungan dari ketiga lokasi pengujian terdiri atas 10.272 data PM2.5, dengan nilai minimum 43 µg/m³, nilai maksimum 612 µg/m³, rata-rata 120,03 µg/m³, median 102 µg/m³, kuartil pertama 82 µg/m³, kuartil ketiga 144 µg/m³, dan standar deviasi 58,40 µg/m³, menunjukkan sebaran kondisi kualitas udara perkotaan yang luas dan representatif. Dataset dibagi menggunakan hold-out split 80:20, menghasilkan 8.217 data latih dan 2.055 data uji, sebagaimana Tabel VIII.

TABEL VIII. PEMBAGIAN DATASET

Subset	Jumlah Data
Data Latih (80%)	8.217
Data Uji (20%)	2.055

Pada dataset uji, Random Forest Regressor memperoleh koefisien determinasi (R^2) sebesar 99,73%, MAE sebesar 1,5464 µg/m³, MSE sebesar 9,4977 µg/m³, dan RMSE sebesar 3,0818 µg/m³, sebagaimana dirangkum pada Tabel IX.

TABEL IX. METRIK PERFORMA RANDOM FOREST REGRESSOR

Metrik	Nilai
R^2 Score	99,73%
Mean Absolute Error (MAE)	1,5464 µg/m³
Root Mean Squared Error (RMSE)	3,0818 µg/m³
Mean Squared Error (MSE)	9,4977 µg/m³

Sebagai perbandingan, penelitian terkait yang memprediksi konsentrasi PM2.5 di wilayah Jakarta menggunakan Random Forest melaporkan MAE sebesar 14,44, RMSE sebesar 18,75, dan R^2 sebesar 0,61 [40], dengan kategorisasi R^2 di atas 0,67 dianggap kuat. Nilai R^2 yang diperoleh pada penelitian ini jauh di atas ambang tersebut dan secara substansial melampaui performa penelitian pembanding, mengindikasikan bahwa model yang dibangun mampu bergeneralisasi dengan baik terhadap dataset multi-lokasi yang digunakan untuk pelatihan dan pengujian.

G. Trade-off Cost dan QoS

TABEL X. TRADE-OFF QOS DAN BIAYA IMPLEMENTASI

Komponen	Manfaat Terukur	Dampak Biaya
Raspberry Pi 4	Inferensi lokal dengan delay 54,84 ms	Harga perangkat lebih tinggi
Firebase RTDB	Packet loss 0,1%, sinkronisasi real-time	Membutuhkan koneksi internet
Wi-Fi	Throughput 45,15 kbps	Membutuhkan infrastruktur jaringan
Random Forest lokal	Prediksi R^2 99,73%	Kebutuhan komputasi lebih tinggi

Menjalankan inferensi secara lokal pada Raspberry Pi 4 menghilangkan ketergantungan pada komputasi berbasis cloud dan mengurangi latensi respons, dengan konsekuensi harga perangkat per unit yang lebih tinggi dibandingkan desain berbasis mikrokontroler saja. Secara keseluruhan, sistem mencapai keseimbangan yang wajar antara kualitas layanan dan biaya implementasi: meskipun lebih mahal dibandingkan sistem monitoring IoT berbasis cloud konvensional, biaya tambahan tersebut sebanding dengan latensi yang lebih rendah, kemampuan pemrosesan lokal, serta layanan monitoring dan prediksi secara real-time.

H. Rangkuman Ketercapaian Spesifikasi

TABEL XI. RANGKUMAN KETERCAPAIAN SPESIFIKASI

Spesifikasi	Realisasi	Status
Monitoring kualitas udara real-time	Data PMS7003, MQ-135, MQ-7, dan DHT22 ditampilkan real-	Tercapai

	time pada AtmoSense	
Pengiriman data sensor ke Firebase	Packet loss 0,1%, delay 54,84 ms, jitter 0,33 ms	Tercapai
Prediksi PM2.5 via Random Forest	R^2 sebesar 99,73%	Tercapai
Tampilan monitoring dan prediksi	Halaman Dashboard, Location, Prediction, dan Device berfungsi sesuai rancangan	Tercapai
Konsumsi daya efisien	Total daya sistem sebesar 17,798 W	Tercapai

Kelima spesifikasi yang ditetapkan pada tahap perancangan tercapai pada tahap implementasi dan pengujian, menunjukkan bahwa arsitektur IoT, edge computing, dan machine learning yang terintegrasi berfungsi sebagaimana dimaksud pada lapisan perangkat keras, komunikasi, prediksi, dan aplikasi.

V. KESIMPULAN

Penelitian ini berhasil mengembangkan sistem monitoring dan prediksi kualitas udara perkotaan berbasis IoT dan *machine learning* yang mengintegrasikan sensor lingkungan, ESP32, Raspberry Pi 4 sebagai *edge device*, Firebase Realtime Database, dan aplikasi AtmoSense. Random Forest Regressor digunakan untuk memprediksi konsentrasi PM2.5 melalui inferensi lokal pada Raspberry Pi 4 dan menghasilkan R^2 sebesar 99,73%, MAE sebesar 1,5464 $\mu g/m^3$, MSE sebesar 9,4977 $\mu g^2/m^6$, dan RMSE sebesar 3,0818 $\mu g/m^3$. Pengujian QoS menghasilkan *throughput* 45,15 kbps, *delay* 54,84 ms, *jitter* 0,33 ms, dan *packet loss* 0,1%, yang termasuk kategori sangat baik berdasarkan standar TIPHON. Seluruh fitur AtmoSense berfungsi sesuai rancangan, sedangkan konsumsi daya keseluruhan sistem sebesar 17,80 W. Hasil tersebut menunjukkan bahwa sistem mampu mendukung monitoring dan prediksi kualitas udara secara terintegrasi dan *real-time*, sehingga seluruh tujuan penelitian berhasil dicapai.

Pengembangan selanjutnya dapat dilakukan dengan memperluas jumlah data, durasi pengujian, dan cakupan lokasi untuk meningkatkan generalisasi model. Algoritma lain seperti XGBoost, LightGBM, Support Vector Regression, dan LSTM juga dapat dibandingkan dengan Random Forest Regressor. Selain itu, sistem dapat dikembangkan dengan fitur notifikasi kualitas udara, pengujian jangka panjang pada lingkungan yang lebih beragam, serta penggunaan sumber energi terbarukan untuk mendukung fleksibilitas implementasi sistem.

REFERENSI

[1] C. Banciu, A. Florea, dan R. Bogdan, "WHO ambient air quality database, 2022 update," 2022.

[2] W. Shi et al., "Edge computing: Vision and challenges," IEEE Internet of Things Journal, vol. 3, no. 5, hlm. 637–646, 2016.

[3] C. Banciu, A. Florea, dan R. Bogdan, "Monitoring and predicting air quality with IoT devices," Processes, vol. 12, no. 9, 2024.

[4] A. Garcia, Y. Saez, I. Harris, X. Huang, dan E. Collado, "Advancements in air quality monitoring: a systematic review of IoT-based air quality monitoring and AI technologies," Artificial Intelligence Review, vol. 58, no. 9, 2025.

[5] M. Elbestar, S. G. Aly, dan R. Ghannam, "Advances in air quality monitoring: A comprehensive review of algorithms for imaging and sensing technologies," 2024.

[6] S. Madakam, R. Ramaswamy, dan S. Tripathi, "Internet of things (IoT): A literature review," Journal of Computer and Communications, vol. 3, no. 5, hlm. 164–173, 2015.

[7] A. H. M. Aman, E. Yadegaridehkordi, Z. S. Attarbashi, R. Hassan, dan Y. J. Park, "A survey on trend and classification of internet of things reviews," 2020.

[8] H. Karnati, "IoT-based air quality monitoring system with machine learning for accurate and real-time data analysis."

[9] M. Mendez, M. G. Merayo, dan M. Nunez, "Machine learning algorithms to forecast air quality: a survey," Artificial Intelligence Review, vol. 56, no. 9, hlm. 10031–10066, 2023.

[10] D. Gil, A. Ferrandez, H. Mora-Mora, dan J. Peral, "Internet of things: A review of surveys based on context aware intelligent services," 2016.

[11] L. Breiman, "Random forests," 2001.

[12] D. N. Azizah, S. Heranurweni, dan L. O. M. Idris, "Internet of things based air quality monitoring system with automatic notification," MALCOM: Indonesian Journal of Machine Learning and Computer Science, vol. 5, no. 3, hlm. 776–787, 2025.

[13] J. Waworundeng, "IoT-based environmental monitoring with data analysis of temperature, humidity, and air quality," COGITO Smart Journal, vol. 10, no. 1, 2024.

[14] A. Dauda, O. Flauzac, dan F. Nolot, "A survey on IoT application architectures," 2024.

[15] A. Choudhary, "Internet of things: a comprehensive overview, architectures, applications, simulation tools, challenges and future directions," 2024.

[16] "Peraturan Menteri Lingkungan Hidup dan Kehutanan Republik Indonesia Nomor P.14/MENLHK/SETJEN/KUM.1/7/2020 tentang Indeks Standar Pencemar Udara," 2020.

[17] W. Shi, J. Cao, Q. Zhang, Y. Li, dan L. Xu, "Edge computing: Vision and challenges," IEEE Internet of Things Journal, vol. 3, no. 5, hlm. 637–646, 2016.

[18] S. S. Gill et al., "Modern computing: Vision and challenges," 2024.

[19] "ESP32 series datasheet version 5.2: 2.4 GHz Wi-Fi + Bluetooth + Bluetooth LE SoC," espressif.com.

[20] R. Sahu et al., "Validation of low-cost sensors in measuring real-time PM10 concentrations at two sites in Delhi National Capital Region," Sensors, vol. 20, no. 5, 2020.

[21] L. Bai et al., "Long-term field evaluation of low-cost particulate matter sensors in Nanjing," Aerosol and Air Quality Research, vol. 20, no. 2, hlm. 242–253, 2020.

[22] M. Lestari, U. Nurbaiti, dan D. Fianti, "The effectiveness of using the MQ-7 sensor integrated Blynk application to detect the existence of gas CO in the air," vol. 17, no. 1, hlm. 76–82, 2021.

[23] "MQ-7 Hanwei datasheet," hwsensor.com.

[24] K. B. K. Sai, S. Ramasubbareddy, dan A. K. Luhach, "IoT based air quality monitoring system using MQ135 and MQ7 with machine learning analysis," Scalable Computing, vol. 20, no. 4, hlm. 599–606, 2019.

[25] "MQ135 Winsen datasheet," winsen-sensor.com, 2015.

[26] W. Adhiwibowo, A. F. Daru, dan A. M. Hirzan, "Temperature and humidity monitoring using DHT22 sensor and Cayenne API," TRANSFORMTIKA, vol. 17, no. 2, hlm. 209–214, 2020.

[27] Y. A. Susetyo, H. A. Parhusip, dan S. Trihandaru, "Herbs go digital: IoT monitors temperature and humidity automatically," COGITO Smart Journal, vol. 10, no. 2, 2024.

[28] A. Hassan et al., "Performance evaluation of Raspberry Pi as an IoT edge signal processing device for a real-time flash flood forecasting system," 2022.

[29] S. E. Mathe, H. K. Kondaveeti, S. Vappangi, S. D. Vanambathina, dan N. K. Kumaravelu, "A comprehensive review on applications of Raspberry Pi," 2024.

[30] "LM2596 SIMPLE SWITCHER power converter 150-kHz 3-A step-down voltage regulator," ti.com, 2023.

[31] K. R. Prasanna, R. Seetharaman, H. M. Lakshmanan, G. Karthik, dan K. Anandan, "Design and fabrication of printed circuit board for IoT applications," 2021.

[32] "Rancang bangun sumber tegangan 12V DC terkendali otomatis menggunakan Arduino Nano."

[33] "Universal Serial Bus Micro-USB cables and connectors specification revision 1.01," 2007.

[34] U. M. Tyas dan A. A. Buckhari, "Implementasi aplikasi Arduino IDE pada mata kuliah sistem digital," 2023.

[35] M. I. Hakiki, U. Darusalam, dan N. D. Nathasia, "Konfigurasi Arduino IDE untuk monitoring pendeteksi suhu dan kelembapan pada ruang data center menggunakan sensor DHT11," Jurnal Media Informatika Budidarma, vol. 4, no. 1, hlm. 150, 2020.

[36] P. Chougale, V. Yadav, A. Gaikwad, dan B. Vidyapeeth, "Firebase – overview and usage," irjmets.com.

[37] C. Khawas dan P. Shah, "Application of Firebase in Android app development – A study," 2018.

[38] U. Patkar, P. Singh, H. Panse, S. Bhavsar, dan C. Pandey, "Python for web development," International Journal of Computer Science and Mobile Computing, vol. 11, no. 4, hlm. 36–48, 2022.

[39] M. Ilhamsyah dan M. Fathurrahman, "Rancang bangun aplikasi Android MySmartFishFeeding menggunakan FlutterFlow dan Firebase Firestore," Jurnal Informatika dan Teknik Elektro Terapan, vol. 12, no. 3S1, 2024.

[40] M. Naufal, A. A. Arsy, dan A. M. Yasir, "PM2.5 concentration prediction model in Jakarta area using Random Forest algorithm," 2025.