

MODUL TERMOSTAT INTERNET OF THINGS BERBASIS PROTOKOL MATTER

Lincoln Abraham, Petrus Santoso
Program Studi Teknik Elektro, Universitas Kristen Petra
Jl. Siwalankerto No. 121-131, Surabaya, 60236, Indonesia
E-Mail: lincolnoct17@gmail.com, petrus@petra.ac.id

Abstrak – Salah satu perangkat rumah yang paling umum adalah AC (Air Conditioner). Seiring perkembangan pesat IoT (Internet of Things) telah muncul banyak termostat. Namun, pertumbuhan ini telah menyebabkan ekosistem yang terfragmentasi, di mana terdapat termostat dari berbagai produsen seringkali bergantung pada protokol komunikasi khusus, sehingga membuat interoperabilitas menjadi tantangan utama. Matter dikembangkan oleh Connectivity Standards Alliance (CSA) untuk mengatasi masalah interoperabilitas ini dengan menyediakan kerangka komunikasi terpadu berdasarkan IPv6 dan teknologi jaringan standar. Dengan menerapkan Matter pada termostat, produsen dapat memastikan integrasi yang mulus di berbagai platform seperti Apple HomeKit, Google Home, dan Amazon Alexa. Penerapan Matter pada termostat dilakukan menggunakan mikrokontroler ESP32 dengan SDK ESP-Matter pada ESP-IDF. Perangkat sudah dicoba dihubungkan dan dikendalikan oleh Home Assistant dan Amazon Alexa. Pada percobaan, perangkat berhasil dihubungkan dengan endpoint AC, sensor suhu, dan sensor kelembapan dikenali oleh aplikasi dan bisa digunakan.

Kata Kunci – internet of things, protokol matter, esp, termostat, interoperabilitas

I. PENDAHULUAN

Internet of Things (IoT) adalah salah satu teknologi yang berkembang dengan pesat dalam 10 tahun terakhir. Teknologi ini menghubungkan berbagai perangkat melalui internet sehingga memungkinkan perangkat-perangkat tersebut untuk saling berkomunikasi dan bertukar data. Teknologi ini telah banyak diadopsi, karena potensinya untuk meningkatkan efisiensi, otomatisasi, dan pengalaman pengguna di berbagai sektor, termasuk hunian pribadi.

Salah satu tantangan terbesar dalam implementasi IoT adalah interoperabilitas antar merk [1], yaitu kemampuan perangkat dari berbagai produsen untuk saling terhubung dan berkomunikasi secara mulus dalam satu ekosistem. Saat ini, perangkat yang dibuat oleh satu perusahaan sering kali tidak kompatibel dengan perangkat dari perusahaan lain. Ketidakterpaduan ini menyulitkan pengguna untuk menciptakan ekosistem IoT yang terintegrasi sepenuhnya. Dalam jangka panjang, tantangan ini dapat menghambat adopsi IoT secara luas, karena pengguna mungkin enggan berinvestasi pada teknologi yang tidak kompatibel dan sulit diperluas.

Protokol Matter merupakan salah satu solusi untuk mengatasi masalah interoperabilitas antar merk dalam ekosistem IoT [2]. Matter dikembangkan oleh Connectivity Standards Alliance (CSA) dengan dukungan dari perusahaan teknologi besar seperti Apple, Google, Amazon, dan lainnya, bertujuan untuk menciptakan standar komunikasi universal yang memungkinkan perangkat pintar dari berbagai merk untuk bekerja secara mulus dalam satu ekosistem. Protokol ini

menggunakan teknologi berbasis IP (Internet Protocol) sehingga kompatibel dengan jaringan yang sudah ada, seperti Wi-Fi dan Thread. Dengan Matter, perangkat dapat terhubung tanpa bergantung pada platform spesifik dari vendor tertentu, yang membuat pengguna dapat lebih fleksibel dalam memilih perangkat untuk rumah pintar mereka tanpa khawatir masalah kompatibilitas.

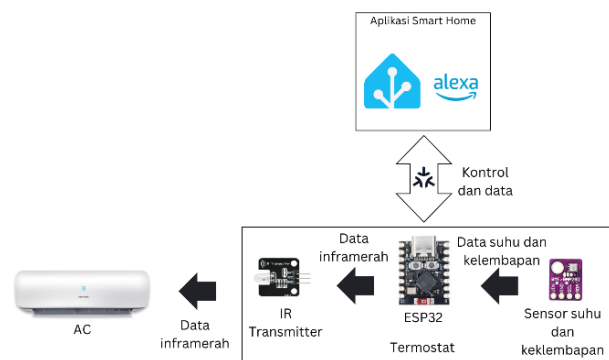
Salah satu mikrokontroler paling populer dan serbaguna untuk IoT adalah ESP32 [3]. ESP32 diprogram menggunakan platform ESP-IDF yang menyediakan integrasi dengan Matter melalui SDK ESP-Matter [4]. ESP32 memiliki varian yang mendukung pengembangan Matter baik yang menggunakan WiFi maupun jaringan Thread.

Suhu dan kelembapan merupakan salah satu faktor yang mempengaruhi kenyamanan dalam hunian, salah satu perangkat yang paling sering digunakan dalam rumah di Indonesia adalah AC. AC dapat dengan mudah dikendalikan melalui sebuah remot. Dengan menggabungkan kendali AC dengan sensor suhu dan kelembapan, pengguna dapat mengotomasi control suhu dan kelembapan dalam rumah. Pada tahun 2022, telah dikembangkan perangkat serupa berbasis ESP8266 dan sensor DHT11 yang menggunakan komunikasi berbasis Blynk [5]. Pengembangan perangkat yang dibuat terletak pada pemanfaatan protokol Matter yang memudahkan integrasi dengan ekosistem yang sudah ada.

II. METODOLOGI

Pengembangan termostat IoT berbasis Matter mencakup desain arsitektur system, desain perangkat keras, pemrograman Matter, serta integrasi dengan aplikasi smart home Home Assistant.

A. Arsitektur Sistem

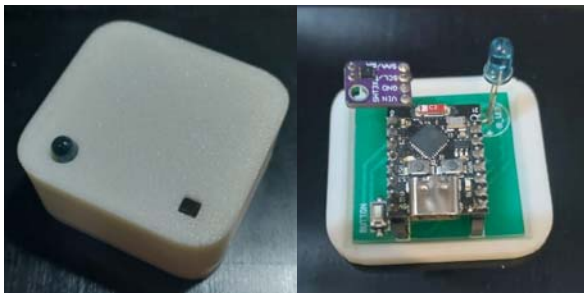


Gambar 1. Arsitektur Sistem Termostat

Sistem ini terdiri dari 3 bagian utama. Yang pertama adalah AC yang akan mengatur suhu ruangan dan kelembapan ruangan. Yang kedua adalah termostat yang akan memantau suhu dan kelembapan dan mengirim perintah kepada AC melalui sinyal inframerah. Bagian ketiga adalah aplikasi smart home yang mengirimkan perintah pada termostat berdasarkan data dari sensor suhu dan kelembapan. Aplikasi berkomunikasi dengan ESP melalui protokol Matter.

B. Perangkat Keras

Termostat terdiri atas ESP32 C3 super-mini, LED inframerah dan sensor SHT31. ESP32 C3 super-mini dipilih karena ukurannya yang kecil sehingga memungkinkan desain kemasan yang kecil. LED inframerah mengirim sinyal inframerah yang dimodulasi membawa perintah sesuai dengan format yang digunakan oleh AC. Sensor suhu yang digunakan adalah SHT31 yang akan berkomunikasi dengan ESP melalui I2C. Gambar 2 menunjukkan implementasi kemasan dan rangkaian termostat.



Gambar 2. Implementasi Termostat

C. Arsitektur Matter

Dalam arsitektur Matter, sebuah perangkat diwakili oleh 1 node. Dalam 1 node bisa terdapat beberapa endpoint yang mewakili fungsi yang terdapat dalam node tersebut. Dalam sebuah endpoint bisa terdapat beberapa cluster yang menyimpan kumpulan attribute yang merupakan data poin. Endpoint mewakili jenis perangkat seperti lampu, saklar, sensor, atau aktuator seperti pompa dan AC. Cluster merupakan kumpulan fungsionalitas yang didukung di dalam Endpoint seperti fungsi on/off pada endpoint lampu. Attribute merupakan variabel-variabel yang menyimpan data yang digunakan dalam sebuah cluster.

Pada node AC terdapat endpoint thermostat yang digunakan untuk mengendalikan suhu dan kelembapan. Endpoint on off light merupakan endpoint kendali biner yang mengatur apakah AC menyala atau mati. Temperature measurement endpoint dan humidity measurement endpoint digunakan untuk menampilkan hasil pengukuran suhu dan kelembapan.

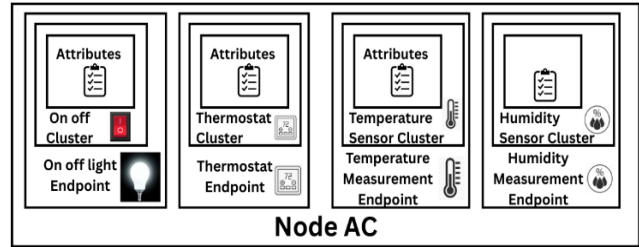
D. Pemrograman ESP-Matter

Untuk melakukan pemrograman Matter pada ESP32 dibutuhkan beberapa persiapan perangkat lunak yang akan digunakan sebagai berikut:

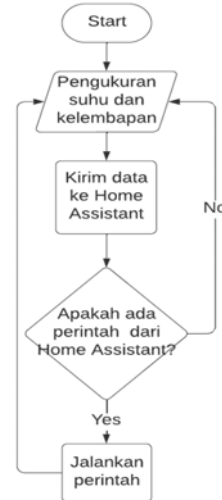
1. Perangkat Host (Linux, Mac OS-X, Windows (WSL))
2. ESP-IDF
3. Matter repository

Gambar 4 merupakan diagram alir dari sistem thermostat. Cara kerja thermostat dimulai pada saat dinyalakan. Pengukuran suhu dan kelembapan akan dilakukan secara berkala menggunakan sensor SHT31. Setelah pengukuran, data akan dikirimkan ke aplikasi smart home untuk ditampilkan dan

disimpan. Perangkat akan bersiaga menerima perintah dari aplikasi.



Gambar 3. Arsitektur Matter Termostat



Gambar 4. Diagram Alir Termostat



Gambar 5. Diagram Alir Pengiriman Data

Gambar 5 merupakan diagram alir pengiriman data dari ESP32 ke aplikasi smart home. ESP32 akan memberikan perintah ke sensor SHT31 untuk melakukan pengukuran. Setelah pengukuran dilakukan Sensor akan mengirimkan hasilnya ke ESP32 dimana data akan dikonversi agar sesuai dengan format yang digunakan pada aplikasi. Selanjutnya ESP32 akan mengirimkan pesan melalui Matter bahwa terdapat perubahan nilai pada atribut suhu dan kelembapan sesuai dengan hasil konversi. Aplikasi akan menerima pesan tersebut dan merubah data data yang ditampilkan serta memasukkan data ke dalam database.



Gambar 6. Diagram Alir Menerima Perintah

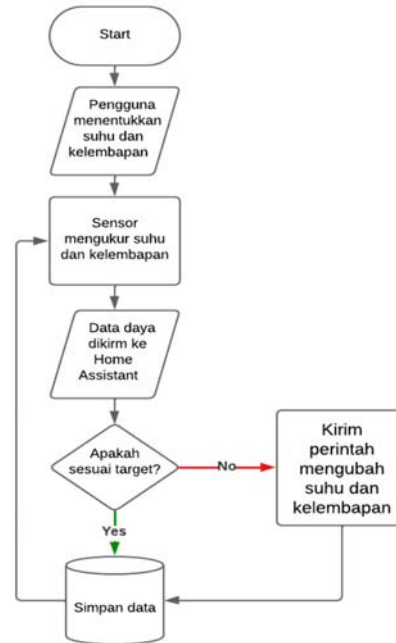


Gambar 7. Diagram Alir Pengiriman Data Melalui Inframerah

Matter menghubungkan ESP32 dengan aplikasi smart home melalui node yang mewakili ESP32. Node akan menerima perintah dan melakukan callback function yang sesuai untuk menjalankan perintah. Perintah berupa perubahan nilai attribute pada aplikasi. ESP32 akan membaca nilai terbaru yaitu suhu, mode, dan power dan mengirimkan perintah yang sesuai melalui LED inframerah.

Gambar 7 merupakan diagram alir pengiriman data menggunakan sinyal inframerah dari LED inframerah yang dikendalikan oleh ESP32. Pada saat ESP dinyalakan, akan dilakukan konfigurasi untuk mempersiapkan driver LEDC pada ESP32. Pengaturan menentukan *clock* yang digunakan serta menyesuaikan resolusi bit dengan kemampuan ESP32 dan frekuensi yang cocok dengan penerima pada AC. Pada saat ESP32 menerima perintah, maka akan dilakukan penyusunan data dengan mengubah variabel sesuai dengan perintah terbaru dan disusun dengan urutan sesuai dengan format yang digunakan AC. *Checksum* dihitung dengan menjumlahkan data yang dikirimkan dan ditambahkan pada akhir data yang akan dikirim. *Checksum* digunakan untuk memverifikasi data yang dikirim untuk memastikan data tidak korup atau hilang pada saat pengiriman. Selanjutnya ESP32 akan melakukan pengiriman dengan mengatur waktu nyala dan mati LED. Data diwakili oleh durasi nyala dan mati LED.

E. Automasi Pada Aplikasi Smart Home



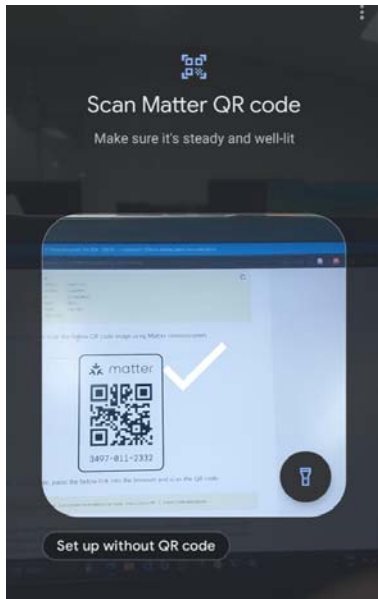
Gambar 8. Diagram Alir Automasi Termostat

Gambar 8 menunjukkan automasi smart termostat. Sistem akan dimulai dari konfigurasi pengguna. Pengguna dapat menentukan suhu dan kelembapan yang nyaman. Selanjutnya, sensor suhu dan kelembapan pada smart termostat akan memonitor suhu dan kelembapan ruangan. Data suhu dan kelembapan dapat diakses oleh pengguna melalui aplikasi atau web Home Assistant. Perintah akan dikirimkan dari aplikasi apabila suhu atau kelembapan tidak sesuai dengan pengaturan pengguna dengan mengubah nilai atribut *power*, suhu, atau mode pada termostat.

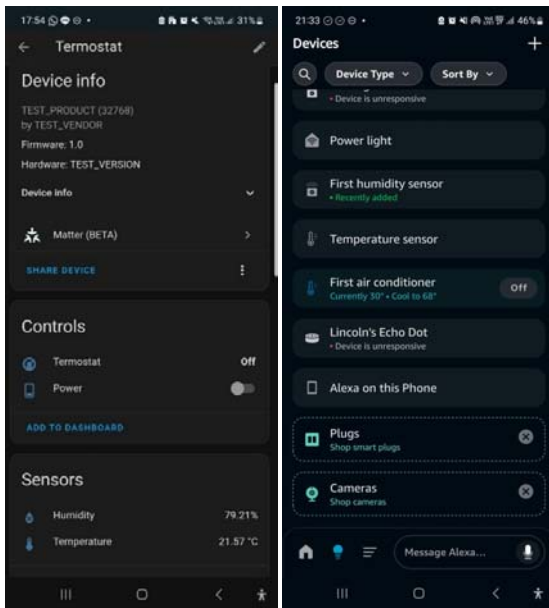
III. PENGUJIAN DAN PEMBAHASAN

Pengujian sistem dilakukan dengan melakukan pengamatan data dan melakukan percobaan menggunakan alat dan aplikasi. Pengujian tersebut diantaranya yaitu menghubungkan perangkat termostat dengan aplikasi Home Assistant. Selanjutnya adalah pengujian akurasi pengukuran konsumsi suhu dan kelembapan ruangan yang akan ditampilkan pada aplikasi. Pengujian terakhir adalah kontrol otomatis termostat berdasarkan pembacaan sensor.

A. Pengujian Menghubungkan Perangkat Ke Aplikasi Smart Home



Gambar 9. Scan QR code Matter



Gambar 10. Perangkat Terhubung Home Assistant(kiri) Amazon Alexa (kanan)

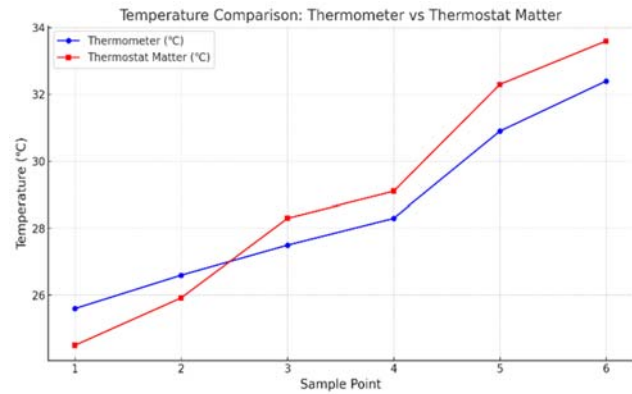
Pengujian ini bertujuan untuk mengetahui apakah termostat dapat terhubung dengan Home Assistant. Pengujian dilakukan dengan menghubungkan termostat dengan aplikasi dan mengecek apakah semua endpoint terdaftar pada aplikasi.

Perangkat dihubungkan pada aplikasi dengan melakukan scan QR code seperti pada gambar 9.

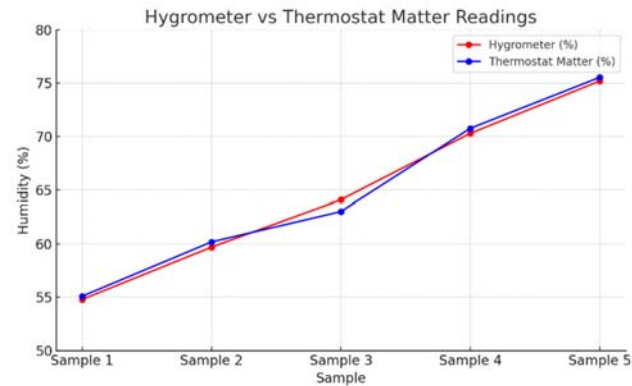
Gambar 10 menunjukkan hasil pengujian menghubungkan perangkat Matter ke aplikasi smart home. Dalam pengujian, perangkat berhasil terhubung dan ditambahkan ke Home Assistant dan Amazon Alexa. Terlihat bahwa endpoint AC, power, humidity, dan temperature telah terdaftar dan bisa digunakan pada aplikasi. Berdasarkan pengujian dibutuhkan hubungan internet yang stabil dan perangkat harus berdekatan dengan smartphone yang melakukan commissioning. Melalui pengujian ini terlihat bahwa dengan menggunakan Matter, perangkat dapat dengan mudah terhubung dengan aplikasi-aplikasi yang ada secara langsung.

B. Pengujian Pengukuran Suhu dan Kelembapan

Pengujian ini dilakukan dengan tujuan untuk mengetahui seberapa akurat pengukuran suhu dan kelembapan oleh termostat. Hal ini dibutuhkan untuk membuat pengaturan suhu dan kelembapan yang lebih sesuai dengan keinginan pengguna.



Gambar 11. Hasil Pengujian Akurasi Pengukuran Suhu



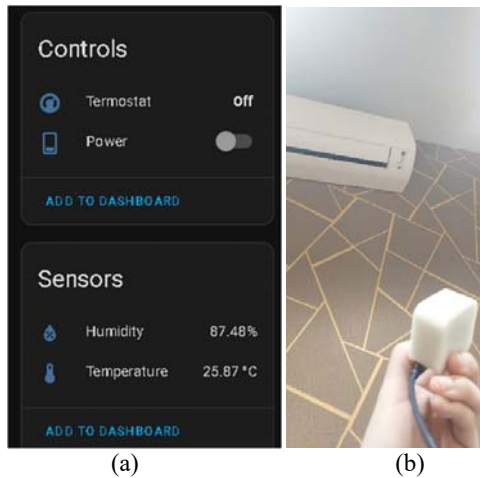
Gambar 12. Hasil Pengujian Akurasi Pengukuran Kelembapan

Gambar 11 dan 12 merupakan grafik hasil pengujian pengukuran suhu dan kelembapan. Dari pengujian pengukuran suhu ditemukan bahwa kesalahan pembacaan terbesar adalah 1.4 derajat Celcius dan persentase terbesar adalah 4.53%. Rata-rata kesalahan pembacaan oleh termostat adalah 3.65%. Selanjutnya pada pengujian pengukuran kelembapan, ditemukan kesalahan pembacaan terbesar sebesar 1.1% kadar air pada udara dengan persentase kesalahan sebesar 1.96%. Rata-rata kesalahan pembacaan sebesar 1.53%. Kesalahan pembacaan yang didapatkan oleh termostat dipengaruhi oleh beberapa hal. Yang pertama adalah inakurasi sensor SHT31 sebesar 1.5% kelembapan ruangan dan 0.1°C [6]. Penyebab

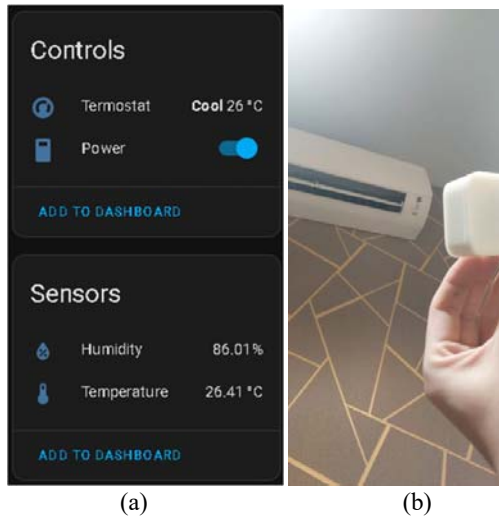
kedua adalah karena desain perangkat di mana sensor diletakan di dalam kemasan sehingga sensor terpengaruh suhu mikrokontroler.

C. Pengujian Pengaturan AC Otomatis

Pengujian ini bertujuan untuk menguji apakah termostat dapat mengendalikan AC secara otomatis berdasarkan desain otomatisasi dari Home Assistant yang disesuaikan dengan suhu ruangan.



Gambar 13. Mematikan AC (a)Home Assistant Mengirim Perintah Off (b) Thermostat Mengirim Perintah Off Ke AC



Gambar 14. Menyalakan AC (a)Home Assistant Mengirim Perintah On (b) Thermostat Mengirim Perintah On Ke AC

Gambar 13 dan 14 merupakan pengujian mengendalikan AC secara otomatis berdasarkan suhu ruangan. Dalam pengujian ditentukan untuk mematikan AC pada saat suhu di bawah 26°C dan menyalakan AC saat suhu berada di atas 26.4°C. AC berhasil dimatikan dan dinyalakan sesuai dengan pengaturan automasi apabila termostat berjarak kurang dari 2 meter. Hal ini disebabkan inakurasi pengaturan waktu on/off LED pada ESP32 serta LED inframerah yang tidak cukup kuat yang menyebabkan data yang diterima pada AC tidak sesuai dengan protokol AC.

IV. KESIMPULAN

Berdasarkan percobaan yang telah dilakukan, perangkat mampu terhubung dengan Home Assistant. Protokol Matter berhasil diintegrasikan dengan ESP32 menggunakan SDK ESP-IDF dan ESP-Matter. Perangkat dapat dengan mudah diintegrasikan pada aplikasi smart home dengan scan QR code. Perangkat Matter dapat langsung dikenali dan didaftarkan beserta semua fiturnya pada aplikasi smart home. Melalui Matter pengembangan perangkat IoT menjadi lebih mudah dikarenakan standarisasi yang diadopsi berbagai ekosistem yang memudahkan setiap perangkat bekerja pada ekosistem-ekosistem tersebut.

DAFTAR PUSTAKA

- [1] Noura, M., Atiquzzaman, M., & Gaedke, M. "Interoperability in internet of things: taxonomies and open Challenges". *Mobile Networks and Applications*, 24(3), 796–809. 2019. Available at: <https://doi.org/10.1007/s11036-018-1089-9>
- [2] Mota, A., Seródio, C., & Valente, A. "Matter protocol integration using espressif's solutions to achieve smart home interoperability". *Electronics*, 13(11), 2217. 2024. Available at: <https://doi.org/10.3390/electronics13112217>
- [3] Maier, A., Sharp, A., & Vagapov, Y. "Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things". *2017 Internet Technologies and Applications (ITA)*, 143–148. 2017. Available at: <https://doi.org/10.1109/ITECHA.2017.8101926>
- [4] Espressif. *Get Started - ESP32 - ESP-IDF Programming Guide v5.4.1 documentation*. 2025. <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/get-started/index.html>
- [5] Kemala, A. P., Syahputra, M. E., Lucky, H., & Achmad, S. "Pengembangan Smart Air Condition Control Menggunakan Platform Blynk Berbasis Mikrokontroler ESP8266 dan Sensor DHT11". *Engineering, Mathematics and Computer Science (EMACS) Journal*, 4(1), 19–23. 2025. Available at: <https://doi.org/10.21512/emacsjournal.v4i1.8072>
- [6] Sensirion. *Datasheet SHT3x*. 2022. Available at: www.sensirion.com