

EVALUASI TENGAH SEMESTER

MATA KULIAH TEKNOLOGI IoT

Dosen : Ahmad Radhy, S.SI., M.SI.

“Sistem IoT Monitoring Suhu Refrigerator Penyimpanan Obat Menggunakan Sensor DS18B20 dan ESP32 pada Platform ThingsBoard”



Disusun Oleh :

Ishmatu Aulia Rizky Kirana (2042231004)

Riska Hidayati Laena (2042231054)

PRODI D4 TEKNOLOGI REKAYASA INSTRUMENTASI

DEPARTEMEN TEKNIK INSTRUMENTASI

FAKULTAS VOKASI

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

2025

DAFTAR ISI

BAB I PENDAHULUAN	2
1.1 Latar Belakang	2
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Manfaat Penelitian	3
BAB II TINJAUAN PUSTAKA	5
2.1 State of The Art	5
2.2 Sensor DS18B20	12
2.3 ESP32-S3	12
2.4 ThingsBoard Cloud	13
2.5 GNUPlot	13
2.6 Rust	14
BAB III METODOLOGI	15
3.1 Komponen	15
3.2 Diagram Alir	15
3.3 Arsitektur Sistem	16
3.4 Wiring Hardware	17
BAB IV HASIL DAN PEMBAHASAN	19
4.1 Hasil Flashing ESP32-S3 dan Pembacaan Sensor DS18B20	19
4.2 Tampilan Hasil Pembacaan Sensor DS18B20 pada Thingsboard	20
4.3 Over The Air (OTA)	21
4.4 Hasil Gnuplot	22
4.5 Hasil Latency	23
BAB V SARAN DAN KESIMPULAN	25
5.1 Kesimpulan	25
5.2 Saran	25

BAB I

PENDAHULUAN

1.1 Latar Belakang

Penyimpanan obat-obatan pada fasilitas kesehatan merupakan aspek yang sangat krusial karena kualitas dan efektivitas obat sangat dipengaruhi oleh kondisi suhu selama penyimpanan. Beberapa jenis obat, seperti vaksin dan antibiotik cair, harus disimpan pada suhu tertentu agar tidak mengalami degradasi kimiawi yang dapat menurunkan efektivitasnya. Namun, dalam praktik di lapangan, pemantauan suhu refrigerator penyimpanan obat masih banyak dilakukan secara manual menggunakan termometer konvensional. Metode ini berpotensi menimbulkan keterlambatan dalam mendeteksi perubahan suhu serta meningkatkan risiko penurunan mutu obat (DAVID SAMUEL BHATTI¹, 2022).

Seiring dengan perkembangan teknologi Internet of Things (IoT), sistem pemantauan suhu kini dapat dilakukan secara otomatis, real-time, dan terintegrasi dengan jaringan internet. IoT memungkinkan berbagai perangkat sensor mengirimkan data suhu secara berkala dan akurat ke platform pemantauan berbasis web. Menurut Budijono dan Felita (Santoso Budijono, 2021), penggunaan mikrokontroler ESP32 yang terintegrasi dengan sensor DS18B20 terbukti efektif dalam memantau perubahan suhu karena sensor ini memiliki tingkat presisi tinggi, konsumsi daya rendah, serta kemudahan integrasi dengan sistem digital. Data suhu yang dikirimkan ke server kemudian dapat divisualisasikan melalui dashboard, sehingga pengguna dapat memantau kondisi suhu refrigerator tanpa harus melakukan pengecekan langsung di lapangan.

Penelitian lain oleh Suwardono (Agus Suwardono, 2024) juga menunjukkan bahwa implementasi sensor DS18B20 dengan mikrokontroler ESP32 mampu melakukan pembacaan suhu secara akurat dan menampilkan hasilnya melalui sistem IoT. Meskipun penelitian tersebut diterapkan pada pemantauan suhu kolam ikan, prinsip dan metode yang digunakan dapat diaplikasikan pada sistem pendingin seperti refrigerator penyimpanan obat. Dengan penerapan teknologi ini, petugas dapat memantau kondisi suhu secara real-time, menyimpan data historis, serta mencegah keterlambatan dalam deteksi perubahan suhu yang dapat berdampak pada kualitas obat.

Berdasarkan uraian tersebut, dapat disimpulkan bahwa penerapan sistem pemantauan suhu berbasis IoT dengan memanfaatkan ESP32 dan sensor DS18B20 yang diintegrasikan dengan

platform ThingsBoard merupakan solusi yang efektif dan efisien untuk meningkatkan keandalan penyimpanan obat di fasilitas kesehatan. Sistem ini memungkinkan proses pemantauan dilakukan secara otomatis dan terpusat, sehingga kualitas obat dapat tetap terjaga serta mendukung terciptanya pengelolaan fasilitas kesehatan yang cerdas dan modern

1.2 Rumusan Masalah

1. Bagaimana merancang sistem Internet of Things (IoT) yang dapat memantau suhu refrigerator penyimpanan obat secara otomatis dan real-time menggunakan sensor DS18B20 dan mikrokontroler ESP32?
2. Bagaimana mengintegrasikan sistem pemantauan suhu tersebut dengan platform ThingsBoard agar data suhu dapat ditampilkan secara visual melalui dashboard monitoring?

1.3 Tujuan Penelitian

1. Merancang dan mengimplementasikan sistem Internet of Things (IoT) berbasis sensor DS18B20 dan mikrokontroler ESP32 untuk melakukan pemantauan suhu refrigerator penyimpanan obat secara otomatis dan real-time
2. Mengintegrasikan sistem pemantauan suhu dengan platform ThingsBoard agar data suhu yang diperoleh dapat ditransmisikan, disimpan, dan divisualisasikan dalam bentuk dashboard interaktif untuk memudahkan proses monitoring
3. Mengevaluasi kinerja sistem IoT yang dirancang dalam meningkatkan efisiensi, akurasi, dan keandalan pemantauan suhu refrigerator, serta membandingkannya dengan metode manual yang masih digunakan di fasilitas kesehatan.

1.4 Manfaat Penelitian

1. Memberikan solusi otomatis dan efisien dalam pemantauan kondisi lingkungan penyimpanan bahan makanan berbasis teknologi IoT.
2. Mendukung upaya pengendalian kualitas bahan makanan dengan menjaga suhu dan kelembaban sesuai standar yang direkomendasikan.

3. Menjadi referensi pengembangan sistem monitoring IoT yang dapat diadaptasi untuk sektor lain seperti pertanian, logistik, dan farmasi.

BAB II

TINJAUAN PUSTAKA

2.1 State of The Art

1. "Attack Scenarios and Security Analysis of MQTT Communication Protocol in IoT System" - Syaiful Andy, Budi Rahardjob, Bagus Hanindhito (2017) Metode Penelitian: Studi ini menganalisis beberapa alasan mengapa banyak sistem Internet of Things (IoT) tidak menerapkan mekanisme keamanan yang memadai untuk protokol MQTT. Penelitian ini kemudian mendemonstrasikan dan menganalisis bagaimana protokol ini dapat diserang dengan mudah menggunakan beberapa skenario serangan. Hasil Penelitian: Hasil penelitian menunjukkan bahwa protokol MQTT secara default hanya menyediakan autentikasi dan tidak mengenkripsi data yang sedang ditransfer, yang menimbulkan masalah pada privasi data, autentikasi, dan integritas data. Setelah kerentanan protokol ini diperiksa, disimpulkan bahwa pengguna dapat
2. "Data Visualization for Wireless Sensor Networks Using ThingsBoard" - Matthew Henschke, Xinzhou Wei, Xiaowen Zhang (2020) Metode Penelitian: Penelitian ini menyajikan proposal visualisasi data untuk Wireless Sensor Networks (WSNs). Metode yang diusulkan menggunakan perangkat keras dan perangkat lunak open source untuk pengumpulan data pada node WSN. Data yang dikumpulkan kemudian divisualisasikan menggunakan platform IoT online yang dikenal sebagai ThingsBoard. Hasil Penelitian: Studi ini berhasil menyajikan skema visualisasi data yang memungkinkan data yang dikumpulkan oleh node dengan sensor berbeda, seperti suhu dan kelembaban, dapat divisualisasikan. Dengan data ini, upaya dapat dilakukan untuk meningkatkan kenyamanan orang di lingkungan tersebut. WSN memainkan peran penting dalam komposisi sistem IoT.
3. "Design of portable refrigerator based on DS18B20 temperature sensor" - Hongxia XU, Weike Wang, Weiwei Deng (2023) Metode Penelitian: Desain ini menggunakan mikrokontroler STC89C52RC dan pendingin semikonduktor untuk merancang kulkas portabel. Desain ini menggunakan modul sensor suhu DS18B20, modul layar SSD1306 OLED, relai, dan heat sink aluminium. Kontrol sakelar radiator berpendingin udara dan sakelar sirip pendingin semikonduktor didasarkan pada prinsip daya lemah mengendalikan daya kuat. Hasil Penelitian: Desain ini merealisasikan fungsi kulkas portabel untuk mengontrol pendingin dan menampilkan suhu secara individual. Modul pengukuran suhu akan menguji suhu real-time di dalam kulkas portabel setelah dihidupkan. Kulkas portabel yang menggunakan lembar pendingin semikonduktor memiliki struktur pendinginan sederhana, tidak menghasilkan polutan, dan dapat mengurangi suhu barang dalam waktu singkat di luar ruangan

4. "MQTT Protocol and Implementation of Equipment Management System for Industrial Internet of Things" - Weiping Wang, Yiling Zhao, Yuan Liu, Guanyin Liu, Fuquan Zheng, Changqing Sun (2024) Metode Penelitian: Penelitian ini berfokus pada protokol MQTT, mekanisme kerjanya, struktur pesan, dan layanan topik. Paket protokol MQTT ditangkap menggunakan alat Wireshark untuk membandingkan dan memverifikasi konsistensi hasil dengan analisis paket protokol. Kode PHP ditulis untuk berlangganan file JSON yang sesuai, membaca, dan menyimpan data dalam basis data yang dirancang. Hasil Penelitian: Studi ini berhasil merancang sistem manajemen peralatan platform cloud Industrial Internet of Things yang mencakup modul penyimpanan data dan tampilan data. Sistem ini menyajikan data yang dikumpulkan oleh perangkat secara visual agar pengunjung dapat melihat data kapan saja, sehingga mewujudkan transmisi dan pemrosesan data yang efisien. Selain itu, pemahaman tentang mekanisme kerja protokol MQTT juga berhasil direalisasikan
5. "A Novel Lo-Ra-Based Wireless Communication Module for Industrial Data Acquisition-Telemetry" - RANU WIJAYA, AHMAD RADHY, AND SAFIRA F. MUJIYANTI (2025) Metode Penelitian: Penelitian ini bertujuan untuk merancang dan mengimplementasikan modul komunikasi Ultra High Frequency (UHF) yang dapat diintegrasikan dengan pemancar suhu konvensional dan pengontrol industri Siemens untuk pengukuran dan pemantauan jarak jauh. Pengujian dan evaluasi dilakukan untuk menilai kinerja modul, termasuk variasi pengukuran dan pengujian jarak antar sistem. Uji lapangan dan evaluasi dilakukan di lingkungan yang terkontrol (Laboratorium Instrumentasi ITS) di mana ia mentransmisikan data suhu Resistance Temperature Detector (RTD). Hasil Penelitian: Temuan utama menunjukkan bahwa sistem mencapai komunikasi nirkabel yang andal hingga 150 meter. Hasilnya menunjukkan bit error rate (BER) sebesar 1.2×10^{-3} , signal-to-noise ratio (SNR) di atas 14 dB, dan packet loss di bawah 2%. Hasil ini mengonfirmasi keandalan dan akurasi modul untuk aplikasi telemetri industri. Solusi yang diusulkan menawarkan platform komunikasi yang hemat biaya dan skalabel yang cocok untuk berbagai lingkungan industri.
6. "Scalable Smart Home Management with ESP32-S3: A Low-Cost Solution for Accessible Home Automation" - Nabil Litayem (2024) Metode Penelitian: Makalah ini menyajikan sistem kendali jarak jauh universal yang terhubung ke internet dan hemat biaya, yang didasarkan pada System on Chip (SoC) ESP32-S3. Sistem yang diusulkan mampu mengendalikan berbagai perangkat rumah (seperti TV, pendingin udara, peralatan pintar, dan pencahayaan) melalui smartphone atau tablet. Sistem ini memanfaatkan kemampuan chip ESP32-S3 untuk menawarkan solusi yang efisien untuk otomatisasi rumah. Hasil Penelitian: Sistem yang diusulkan mengintegrasikan fitur-fitur kompetitif seperti efektivitas biaya, daya rendah, komunikasi nirkabel berkinerja tinggi (Wi-Fi dan Blue-

tooth), dan tingkat skalabilitas yang tinggi. Solusi ini menawarkan sistem yang efisien untuk otomatisasi rumah. Makalah ini juga menyoroti pentingnya antarmuka kontrol yang intuitif dan mudah diakses untuk meningkatkan otonomi pengguna, terutama bagi penyandang disabilitas fisik

7. "Monitoring the Smart City Sensor Data Using Thingsboard and Node-Red" - Elham Okhovat, Michael Bauer (2021) Metode Penelitian: Penelitian ini memperkenalkan arsitektur yang bertujuan untuk mengumpulkan, memvisualisasikan, dan memantau aliran data dari sensor dan komponen infrastruktur. Platform IoT ThingsBoard digunakan untuk pengumpulan dan visualisasi data. Node-Red digunakan untuk mengkategorikan data berdasarkan nama sensor. Proses ini diilustrasikan dengan mensimulasikan pengumpulan data dalam lingkungan smart city yang disimulasikan. Hasil Penelitian: Arsitektur yang diusulkan memungkinkan pengumpulan data tentang lingkungan operasional dan mengilustrasikan bagaimana pengumpulan dan visualisasi data dapat diintegrasikan. Eksperimen menggunakan set data sensor disediakan untuk mengilustrasikan pendekatan, pemrosesan, dan visualisasi. Tujuan akhirnya adalah untuk memvisualisasikan data melalui dasbor ThingsBoard, memberikan otoritas manajemen status saat ini dari semua data sensorik, dan memicu alarm sesuai dengan status data yang telah ditentukan
8. "An Efficient OTA firmware updating Architecture based on LoRa suitable for agricultural IoT Applications" - Samy H. Sharf, Rafaat Abd Elhamied, Madonna K. Habib, and Ahmed H. Madian (2021) Metode Penelitian: Penelitian ini mengusulkan arsitektur pembaruan firmware Over-The-Air (OTA) yang andal dan efisien yang memanfaatkan teknologi LoRa untuk mendistribusikan pembaruan firmware dari jarak jauh untuk aplikasi Internet of Things Pertanian (Ag-IoT). Arsitektur ini terdiri dari sistem hardware (sirkuit pengunggah firmware) dan software (aplikasi desktop yang dikembangkan) yang terhubung untuk menyelesaikan prosedur pembaruan firmware OTA. Hasil Penelitian: Kontribusi utama dari penelitian ini adalah keberhasilan mengintegrasikan teknologi LoRa untuk pembaruan firmware OTA, menjadikannya solusi yang sangat cocok untuk aplikasi Ag-IoT. Hal ini memungkinkan pembaruan firmware pada microchip target, seperti ATMEGA8L, secara nirkabel tanpa memerlukan kabel fisik, sehingga meningkatkan efisiensi operasional
9. "Cloud-Based IoT System for Real-Time Harmful Algal Bloom Monitoring: Seamless ThingsBoard Integration via MQTT and REST API" - Ammar Haziq Annas, Nik Nor Muhammad Saifudin Nik Mohd Kamal, Ahmad Anwar Zainuddin, Normawaty Mohammad Noor (2024) Metode Penelitian: Penelitian ini merancang dan mengimplementasikan sistem IoT berbasis cloud untuk pemantauan Harmful Algal Bloom (HAB) secara real-time. Sistem ini menggunakan NodeMCU ESP32-WROOM-32 untuk pengumpulan data sensor dan mengintegrasikannya ke platform ThingsBoard. Integrasi dilakukan

melalui protokol komunikasi MQTT untuk transmisi data real-time dan REST API untuk pengelolaan perangkat dan data. Hasil Penelitian: Sistem yang diusulkan berhasil mengintegrasikan ThingsBoard menggunakan MQTT dan REST API untuk memantau data kualitas air secara real-time. Hasil ini menunjukkan potensi besar sistem tersebut dalam membantu operasi pemantauan HAB, memungkinkan visualisasi data real-time yang cepat serta mendukung pengambilan keputusan yang tepat untuk tindakan konservasi

10. "Embedded Real-Time Human Activity Recognition on an ESP32-S3 Microcontroller Using Ambient Audio Data" - Thomas Pfitzinger, Hendrik Wöhrle (2023) Metode Penelitian: Penelitian ini mengembangkan perangkat sensor cerdas yang melakukan pengenalan aktivitas manusia (Human Activity Recognition/HAR) secara real-time pada mikrokontroler ESP32-S3 menggunakan data audio ambien. Metodenya melibatkan perancangan Convolutional Neural Network (CNN) yang kemudian direduksi ukuran dan biaya komputasinya melalui neural architecture search yang dioptimalkan untuk perangkat embedded dan kuantisasi model, sehingga memungkinkan inferensi pada mikrokontroler. Hasil Penelitian: Model CNN yang dioptimalkan menunjukkan kinerja yang baik, mencapai akurasi klasifikasi rata-rata sebesar 82,2persen untuk data HAR dan akurasi rata-rata 83,5persen untuk data audio ambien. Hasil ini menunjukkan bahwa pengenalan aktivitas manusia berbasis audio real-time yang efisien dapat dilakukan secara langsung pada perangkat embedded berdaya rendah, mengurangi lalu lintas jaringan dan meningkatkan privasi.
11. "ESP32-S3 DevKit-C Integration for Enhanced Road Safety For Pothole Detection and Nighttime Driving Assistance" - Manas Mishra, Akanksha Shukla, Hina Varshney (2024) Metode Penelitian: Penelitian ini memanfaatkan papan ESP32-S3 DevKit-C untuk mengembangkan solusi guna meningkatkan keselamatan jalan dengan mengatasi masalah lubang jalan (pothole) dan tantangan mengemudi di malam hari. Fokusnya adalah pada deteksi bahaya jalan yang cepat dan implementasi lampu depan yang cerdas dan penuh pertimbangan. Hasil Penelitian: Sistem yang diusulkan bertujuan untuk mewujudkan deteksi lubang jalan yang cepat dan pencahayaan cerdas di malam hari, yang akan membantu mengurangi frekuensi kecelakaan yang berasal dari lubang tersembunyi dan lampu depan yang menyilaukan. Penelitian ini memberikan dasar untuk sistem bantuan mengemudi yang lebih cerdas dan proaktif untuk keselamatan jalan
12. "FireEye: An IoT-Based Fire Alarm and Detection System for Enhanced Safety" - J Bino, Muhammad Fiazul Haque, Md. Moynul Islam, Udaya Mouni Boppana, Nahida Fatme, Md AL Mahbub Hossain, C Manjula (2024) Metode Penelitian: Penelitian ini berfokus pada perancangan dan implementasi sistem alarm dan deteksi kebakaran berbasis IoT bernama FireEye. Sistem ini menggunakan sensor yang terhubung ke internet

dan mampu mendeteksi api, asap, dan gas berbahaya secara real-time untuk memberikan peringatan dini kepada pengguna melalui notifikasi seluler. Hasil Penelitian: Sistem Fire-Eye yang diusulkan berhasil meningkatkan kemampuan deteksi kebakaran yang cepat dan akurat, yang secara signifikan dapat mempercepat waktu respons dalam situasi darurat dan mengurangi risiko bahaya properti dan nyawa. Solusi ini mewakili pendekatan yang lebih cerdas dan proaktif untuk keselamatan kebakaran rumah tangga maupun industri

13. "MQTT Protocol and Implementation of Equipment Management System for Industrial Internet of Things" - Weiping Wang, Yiling Zhao, Yuan Liu, Guanyin Liu, Fuquan Zheng, Changqing Sun (2024) Metode Penelitian: Penelitian ini berfokus pada analisis protokol MQTT, mekanisme kerja, struktur pesan, dan layanan topiknya untuk mengatasi kesulitan berbagi data di sistem industri yang tertutup. Metode penelitian melibatkan penangkapan paket protokol MQTT menggunakan alat Wireshark untuk verifikasi, penulisan kode PHP untuk berlangganan file JSON dan penyimpanan data, serta perancangan sistem manajemen peralatan cloud IIoT. Hasil Penelitian: Penelitian ini berhasil merancang sistem manajemen peralatan platform cloud Industrial Internet of Things (IIoT) yang terdiri dari modul penyimpanan dan tampilan data. Sistem ini memvisualisasikan data yang dikumpulkan oleh perangkat, sehingga mewujudkan transmisi dan pemrosesan data yang efisien dan dapat diakses secara visual oleh pengunjung
14. "Research on Accurate OTA Testing for Large-Form-Factor 5G IoT Devices in Medium-Sized Anechoic Chamber" - Meijun Qu, Kai Zhang, and Siyang Sun (2022) Metode Penelitian: Penelitian ini mengevaluasi apakah kinerja Over-The-Air (OTA) yang akurat dari perangkat IoT Large-Form-Factor (LFF) 5G dapat diperoleh dari ruang anechoic berukuran sedang, terutama ketika lokasi antena pada perangkat tersebut bergeser secara signifikan dari pusat zona tenang. Sebuah figure of merit baru, ROFF, didefinisikan untuk mengkarakterisasi sejauh mana pergeseran dari pusat, dan pengaruhnya pada kinerja OTA disajikan melalui analisis statistik dan simulasi. Hasil Penelitian: Ditemukan bahwa penyimpangan kinerja ditentukan oleh ROFF (sejauh mana pergeseran dari pusat) daripada jarak pergeseran absolut. Hubungan antara ROFF dan penyimpangan kinerja dapat diturunkan menggunakan metode least-squares fitting. Selain itu, dikonfirmasi bahwa penyimpangan kinerja tidak dapat dihilangkan hanya dengan meningkatkan area pengukuran dalam skenario pergeseran dari pusat
15. "The Reading Data Error Analysis of 1-wire Bus Digital Temperature Sensor DS18B20" - Bangyan Huang, Jianglong Lei and Yan bo (2012) Metode Penelitian: Penelitian ini memberikan prinsip pengukuran suhu dan program untuk sensor suhu digital bus 1-wire DS18B20 dalam sistem mikrokontroler (MCU). Analisis utamanya berfokus pada mekanisme bagaimana kesalahan dihasilkan saat membaca suhu, dan bagaimana solusi

untuk masalah ini dapat diberikan. Penelitian ini menekankan pada pentingnya celah waktu (timing gap) saat membaca perangkat. Hasil Penelitian: Ditemukan bahwa celah waktu saat membaca perangkat sangat penting. Jika celah waktu tidak sesuai dengan panjang waktu yang diberikan dalam dokumen spesifikasi DS18B20, kesalahan akan terjadi secara tak terhindarkan. Metode untuk mendapatkan kesimpulan ini disediakan sebagai referensi yang berguna saat menggunakan platform pengembangan 'Keil C'

16. "The Research on Security Technology of Earthquake Warning Information Release Based on Zero Trust MQTT Protocol" - Luqiang Sun (2023) Metode Penelitian: Penelitian ini membahas masalah keamanan dalam sistem rilis informasi peringatan dini gempa bumi yang menggunakan protokol MQTT, di mana pengguna akhir yang berlangganan informasi sebagian besar adalah pengguna zero trust. Penelitian ini mengusulkan sebuah teknologi keamanan yang ditingkatkan untuk rilis informasi peringatan gempa berbasis prinsip Zero Trust. Hasil Penelitian: Tujuan utama penelitian ini adalah untuk mengatasi masalah keamanan yang timbul dari pengguna zero trust yang berlangganan melalui protokol MQTT, yang merupakan potensi bahaya keamanan terbesar. Solusi yang diusulkan bertujuan untuk meningkatkan keamanan pelepasan informasi peringatan dini gempa bumi yang bergantung pada jaringan Internet
17. "Towards User Friendly Smart Precision Farming: Assessing ThingsBoard as an Interface for IoT Based Farming System Using System Usability Scale" Kadek Darmaastawan, I Wayan Aditya Suranata (2024) Metode Penelitian: Penelitian ini bertujuan untuk mengevaluasi kegunaan (usability) platform ThingsBoard sebagai antarmuka cloud untuk sistem pertanian presisi cerdas berbasis IoT (khususnya Universal Digital Agriculture Watering Assistant - UDAWA). Evaluasi kegunaan dilakukan menggunakan kuesioner System Usability Scale (SUS), dengan fokus pada kelompok petani skala kecil di Indonesia, yang didominasi oleh Generasi X dan Baby Boomers. Hasil Penelitian: Penelitian ini menyajikan hasil evaluasi usability ThingsBoard dalam konteks aplikasi pertanian. Temuan ini penting untuk mengatasi kesenjangan pengetahuan mengenai kegunaan ThingsBoard dalam aplikasi pertanian dan untuk memastikan bahwa antarmuka yang digunakan dalam sistem pertanian cerdas berbasis IoT ramah pengguna, terutama bagi kelompok petani yang menghadapi tantangan adaptasi teknologi.
18. "Intelligent Refrigerator Using Artificial Intelligence" - Shweta A.S (2017) Metode Penelitian: Penelitian ini mengembangkan sebuah Intelligent Refrigerator Module yang dapat diintegrasikan dengan kulkas biasa menggunakan teknologi Artificial Intelligence (AI). Sistem menggunakan kamera mikro 360° untuk memantau isi kulkas, algoritma pengolahan citra untuk mengenali dan menghitung usia sayuran atau buah berdasarkan warna, tekstur, bentuk, dan ukuran, serta prosesor mikro untuk mengirimkan informasi tersebut ke pengguna. Sistem juga mengeluarkan notifikasi suara atau SMS apabila terdapat

bahan makanan yang sudah lama tidak digunakan. Hasil Penelitian: Prototipe kulkas cerdas ini mampu mengidentifikasi jenis sayuran dengan akurasi hingga 96,55persen menggunakan pencocokan histogram warna dan tekstur. Sistem dapat melacak usia bahan makanan dan memberi tahu pengguna apabila ada bahan yang sudah tidak dimasukkan selama 30 hari. Dengan demikian, teknologi ini membantu menjaga kesegaran makanan dan meningkatkan kesadaran nutrisi pengguna secara efisien serta hemat biaya.

19. "Time Series Ordinal Regression for Supporting the Storage of Temperature-Sensitive Medication in Domestic Refrigerators" - Ali Hammadeh, Amin Zayani, Andreas W. Kempa-Liehr (2021) Metode Penelitian: Penelitian ini memanfaatkan time series analysis dan machine learning untuk mendeteksi penyimpanan obat sensitif suhu dalam kulkas rumah tangga. Data diambil dari sensor pintar MedAngel, yang mencatat suhu setiap 3 menit. Data ini diolah menggunakan pustaka tsfresh untuk mengekstraksi 542 fitur deret waktu. Model LightGBM classifier digunakan untuk memprediksi kemungkinan pelanggaran suhu ideal (2°C – 8°C) satu jam ke depan berdasarkan pola suhu lima jam sebelumnya. Hasil Penelitian: Model berhasil memprediksi pelanggaran suhu dengan koefisien korelasi Matthews (MCC) antara 0.43–0.88, tergantung pada variasi kulkas. Fitur paling berpengaruh adalah minimum/maximum temperature, Benford correlation, dan approximate entropy. Hasil menunjukkan bahwa metode time-series ordinal regression dapat secara efektif mendeteksi anomali suhu dan berpotensi meningkatkan keamanan penyimpanan obat termolabil di rumah tangga.
20. "Wireless Temperature Sensor Network Based on DS18B20, CC2420, MCU AT89S52" - Fang Xiong (2018) Metode Penelitian: Penelitian ini merancang jaringan sensor suhu nirkabel (Wireless Temperature Sensor Network – WTSN) berbasis sensor DS18B20, transceiver CC2420, dan mikrokontroler AT89S52 menggunakan protokol komunikasi ZigBee (IEEE 802.15.4). Sistem terdiri atas beberapa node sensor yang mengirimkan data suhu secara berantai ke substation dan kemudian ke komputer pusat. Perangkat lunak dirancang untuk mengatur transmisi data, alarm suhu tinggi/rendah, dan pengaturan batas ambang melalui antarmuka tombol sederhana. Hasil Penelitian: Hasil uji menunjukkan sistem memiliki akurasi tinggi pada rentang suhu -50°C hingga 120°C dengan jarak transmisi andal hingga 100 meter dan tingkat kesalahan rendah (13persen). Sistem ini hemat daya, berbiaya rendah, dan cocok diterapkan pada berbagai bidang Internet of Things (IoT), seperti pemantauan lingkungan gudang, penyimpanan makanan, atau tambang.

2.2 Sensor DS18B20

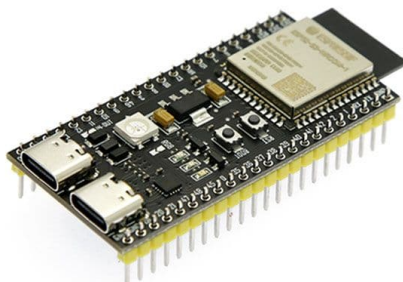


Gambar 1: Sensor DS18B20

Sensor DS18B20 merupakan sensor digital yang berfungsi untuk mengukur suhu dengan akurasi tinggi dan respon yang cepat. Sensor ini memiliki rentang pengukuran dari -55 °C hingga +125 °C dengan akurasi mencapai $\pm 0,5$ °C pada kisaran suhu -10 °C hingga +85 °C. DS18B20 bekerja menggunakan antarmuka 1-Wire, yang memungkinkan komunikasi data hanya dengan satu jalur sinyal, sehingga instalasinya lebih sederhana dan efisien.

Sensor ini memiliki resolusi yang dapat diatur antara 9 hingga 12 bit, setara dengan ketelitian hingga 0,0625 °C, sehingga mampu memberikan hasil pengukuran yang sangat detail. Dengan konsumsi daya yang rendah, ukuran fisik yang kecil, serta kemampuan bekerja stabil dalam berbagai kondisi lingkungan, DS18B20 banyak digunakan pada aplikasi peman-tauan suhu ruang, sistem pendingin, perangkat elektronik, maupun sistem otomasi industri yang memerlukan ketepatan dan keandalan tinggi dalam pengukuran suhu.

2.3 ESP32-S3



Gambar 2: ESP32-S3

Sebagai mikrokontroler utama, ESP32-S3 berperan dalam menyalurkan data hasil pembacaan sensor ke platform ThingsBoard untuk proses pemantauan dan analisis lebih lanjut. Mikrokontroler ini memiliki prosesor dual-core Xtensa LX7 dengan dukungan Wi-Fi dan Bluetooth Low Energy (BLE), sehingga mampu melakukan komunikasi data secara nirkabel dengan efisien dan andal.

Dalam proyek ini, ESP32-S3 menerima data dari sensor secara periodik, kemudian melakukan pemrosesan awal seperti filtering atau kalibrasi, sebelum mengirimkannya ke ThingsBoard melalui protokol MQTT. Dukungan memori yang besar, konsumsi daya rendah, serta kemampuan pemrograman berbasis FreeRTOS menjadikan ESP32-S3 sangat sesuai untuk aplikasi Internet of Things (IoT) yang membutuhkan pengiriman data real-time dan konektivitas stabil antara perangkat lapangan dan server cloud

2.4 ThingsBoard Cloud



Gambar 3: ThingsBoard Cloud

ThingsBoard adalah platform IoT berbasis cloud yang berfungsi untuk mengelola dan menampilkan data sensor dalam bentuk dashboard interaktif. Melalui protokol MQTT, data suhu dan kelembaban yang dikumpulkan dari perangkat dapat dikirim secara real-time ke ThingsBoard untuk kemudian divisualisasikan dalam berbagai bentuk seperti grafik tren, indikator status, maupun tabel data.

Platform ini memungkinkan pengguna untuk melakukan pemantauan kondisi lingkungan secara menyeluruh, serta menyediakan fitur device management, pengaturan notifikasi (alert system), dan otomatisasi aksi berdasarkan parameter tertentu. Dengan fleksibilitas tinggi dan tampilan antarmuka yang intuitif, ThingsBoard menjadi solusi ideal untuk sistem monitoring dan kontrol IoT yang membutuhkan keandalan, skalabilitas, serta kemudahan dalam analisis data

2.5 GNUPlot

Gnuplot merupakan perangkat lunak visualisasi data dan plotting grafik yang digunakan untuk menampilkan hasil pengukuran atau analisis dalam bentuk grafik dua dimensi maupun

tiga dimensi. Dalam proyek ini, Gnuplot berperan untuk menampilkan data suhu dan kelembaban yang tersimpan pada database InfluxDB secara real-time maupun historis, sehingga memudahkan proses analisis tren dan evaluasi performa sistem.

Dengan dukungan berbagai format output seperti PNG, SVG, dan PDF, serta kemampuan scripting yang fleksibel, Gnuplot memungkinkan pengguna untuk melakukan otomatisasi pembuatan grafik sesuai kebutuhan. Selain itu, tampilannya yang ringan dan kompatibilitas lintas platform menjadikan Gnuplot sebagai alat yang efisien dan andal dalam visualisasi data teknik maupun ilmiah

2.6 Rust



Gambar 4: Rust

Rust merupakan bahasa pemrograman tingkat sistem yang dirancang untuk memberikan keamanan memori, performa tinggi, dan efisiensi penggunaan sumber daya. Dalam proyek ini, Rust digunakan untuk memprogram mikrokontroler ESP32-S3, yang berperan dalam membaca data dari sensor serta mengirimkannya ke platform ThingsBoard secara real-time melalui protokol MQTT.

Dengan sistem manajemen memori tanpa garbage collector dan dukungan konkurensi aman, Rust memungkinkan pengembangan perangkat IoT yang stabil, responsif, dan bebas dari bug umum seperti data race atau null pointer error. Selain itu, struktur kode yang modular serta fitur error handling yang kuat menjadikan Rust sangat sesuai untuk pengembangan sistem tertanam (embedded system) yang menuntut keandalan tinggi dan efisiensi eksekusi

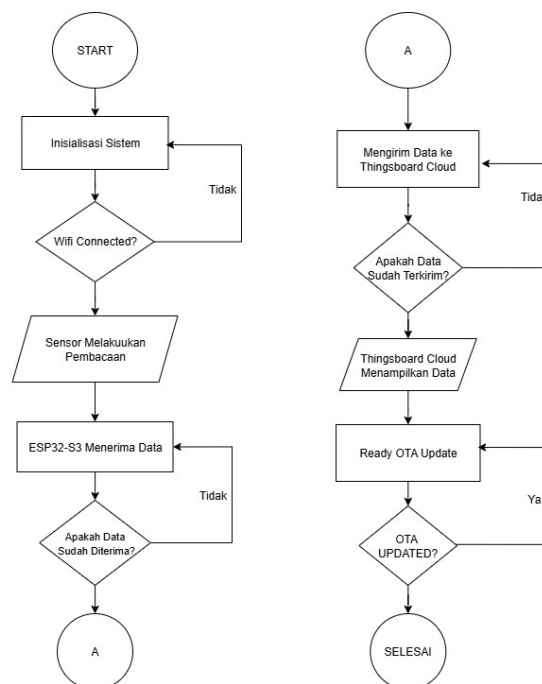
BAB III

METODOLOGI

3.1 Komponen

1. 1x ESP32-S3
2. 1x Sensor DS18B20
3. 3x Kabel Jumper
4. 1x Breadboard

3.2 Diagram Alir



Gambar 5: Diagram Alir system

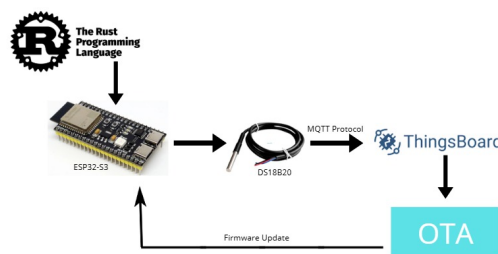
Diagram alir sistem menggambarkan alur kerja keseluruhan proses pembacaan dan pengiriman data suhu dari sensor hingga data tersebut ditampilkan pada platform ThingsBoard Cloud. Proses dimulai ketika sistem diaktifkan (Start). Pada tahap ini, seluruh komponen seperti sensor DS18B20, mikrokontroler ESP32-S3, serta koneksi Wi-Fi diinisialisasi agar siap melakukan pembacaan data. Selanjutnya, sensor DS18B20 berfungsi untuk mendeteksi dan mengukur suhu lingkungan di sekitarnya. Hasil pembacaan suhu kemudian dikirimkan ke mikrokontroler ESP32-S3 untuk diproses lebih lanjut.

ESP32-S3 menerima data dari sensor melalui komunikasi One-Wire. Setelah itu, sistem melakukan pengecekan apakah data telah diterima dengan benar. Jika data belum diterima, maka ESP32-S3 akan mengulangi proses penerimaan hingga data dapat diperoleh secara utuh. Setelah data berhasil diterima, ESP32-S3 akan mengirimkan data suhu ke platform ThingsBoard Cloud menggunakan koneksi Wi-Fi melalui protokol MQTT (Message Queuing Telemetry Transport). Protokol MQTT dipilih karena efisien, ringan, dan sesuai untuk komunikasi data pada sistem Internet of Things (IoT).

Tahapan berikutnya adalah verifikasi pengiriman data. ESP32-S3 akan memastikan bahwa data telah terkirim ke ThingsBoard Cloud. Jika pengiriman gagal, sistem akan melakukan pengiriman ulang hingga data berhasil dikirim. Setelah data diterima oleh ThingsBoard Cloud, platform tersebut menampilkan hasil pembacaan suhu dalam bentuk visual seperti grafik atau indikator nilai suhu secara real-time pada dashboard. Dengan demikian, pengguna dapat memantau kondisi suhu dari jarak jauh secara langsung dan berkesinambungan.

Secara keseluruhan, diagram alir ini menunjukkan bahwa sistem bekerja secara otomatis dan berulang, dimulai dari proses pembacaan sensor, penerimaan serta verifikasi data oleh ESP32-S3, pengiriman ke ThingsBoard Cloud, hingga penampilan data kepada pengguna. Alur ini memastikan bahwa sistem monitoring suhu berbasis IoT dapat berjalan dengan baik, efisien, dan memberikan data yang akurat secara real-time.

3.3 Arsitektur Sistem



Gambar 6: Arsitektur Sistem

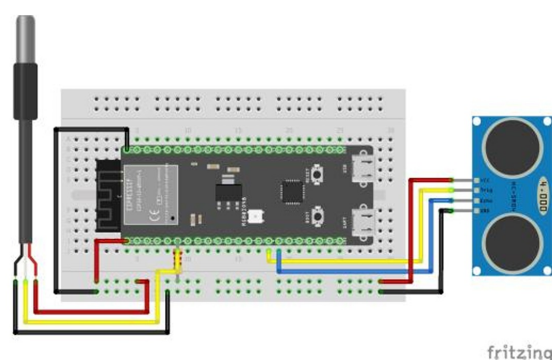
Sistem yang digambarkan pada diagram di atas merupakan implementasi Internet of Things (IoT) berbasis mikrokontroler ESP32-S3 yang diprogram menggunakan bahasa pemrograman Rust. Pemilihan Rust didasari oleh keunggulannya dalam hal keamanan memori, efisiensi eksekusi, serta stabilitas sistem yang tinggi, sehingga sangat sesuai untuk pengembangan firmware perangkat tertanam yang menuntut keandalan. Firmware hasil kompilasi Rust diimplementasikan pada ESP32-S3, yang berfungsi sebagai pusat kendali untuk mengatur seluruh proses mulai dari inisialisasi koneksi jaringan WiFi, pembacaan data dari sensor, hingga pengiriman data ke server cloud.

Sensor DS18B20 berperan sebagai perangkat pengukur suhu digital yang berkomunikasi dengan ESP32-S3 melalui protokol 1-Wire. Sensor ini memiliki tingkat akurasi tinggi dan mampu mendeteksi suhu dalam rentang luas, menjadikannya ideal untuk pemantauan suhu pada aplikasi seperti penyimpanan obat atau lingkungan terkendali. Data hasil pembacaan suhu kemudian dikirimkan oleh ESP32-S3 ke platform ThingsBoard menggunakan protokol MQTT (Message Queuing Telemetry Transport). Protokol ini memungkinkan pertukaran data secara ringan, cepat, dan andal antara perangkat dan server, yang sangat sesuai untuk sistem IoT berskala besar.

Platform ThingsBoard berfungsi sebagai pusat pengelolaan data, penyimpanan, serta visualisasi hasil pengukuran melalui dashboard interaktif. Selain menerima dan menampilkan data suhu secara real-time, ThingsBoard juga mendukung fitur Over-The-Air (OTA) Update yang memungkinkan pembaruan firmware ESP32-S3 dilakukan secara jarak jauh tanpa perlu intervensi fisik. Proses OTA dilakukan dengan mengirimkan perintah pembaruan dari server ke perangkat, di mana ESP32-S3 kemudian mengunduh firmware terbaru melalui koneksi WiFi dan memperbarui sistem secara otomatis.

Secara keseluruhan, sistem ini membentuk ekosistem IoT yang efisien, aman, dan berkelanjutan. Integrasi antara Rust, ESP32-S3, sensor DS18B20, ThingsBoard, dan fitur OTA memungkinkan proses pemantauan suhu berjalan secara otomatis, real-time, serta mudah dipelihara. Dengan kemampuan pembaruan firmware nirkabel, sistem ini tidak hanya menjamin kontinuitas operasional, tetapi juga memastikan perangkat selalu berada pada versi terbaru dengan performa dan keamanan yang optimal.

3.4 Wiring Hardware



Gambar 7: Wiring Hardware

Pada sistem ini digunakan satu buah sensor DS18B20 yang berfungsi untuk mengukur suhu di dalam refrigerator penyimpanan obat secara digital, serta sebuah mikrokontroler ESP32-S3 yang bertugas membaca hasil pengukuran tersebut dan mengirimkannya ke platform ThingsBoard Cloud melalui koneksi Wi-Fi. Seluruh komponen dirangkai di atas breadboard untuk

memudahkan proses perakitan, pengujian, dan pengembangan sistem. Jalur daya (VCC) dan ground (GND) pada breadboard disatukan (common rail) agar seluruh komponen mendapatkan sumber tegangan dan referensi ground yang sama, sehingga rangkaian dapat berfungsi secara stabil dan konsisten.

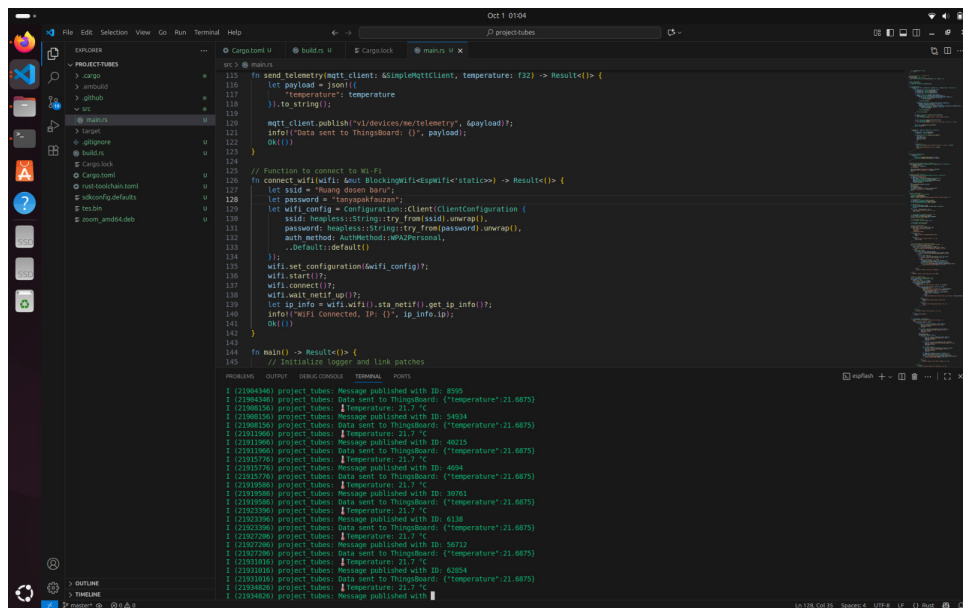
Sensor DS18B20 memiliki tiga jalur utama yaitu VCC, GND, dan Data. Dalam konfigurasi rangkaian ini, kabel merah dari sensor dihubungkan ke pin 3.3V pada ESP32-S3 untuk menyuplai daya, kabel hitam dihubungkan ke pin GND sebagai jalur ground, sedangkan kabel kuning sebagai jalur data dihubungkan ke GPIO 5 pada ESP32-S3. Jalur data ini dilengkapi dengan resistor pull-up sebesar $4.7k\Omega$ yang terhubung antara pin data dan VCC untuk menjaga kestabilan komunikasi One-Wire, sehingga pembacaan suhu dapat dilakukan dengan akurat tanpa gangguan sinyal.

Mikrokontroler ESP32-S3 berperan sebagai unit pengendali utama yang menjalankan program berbasis bahasa Rust. Program ini menginisialisasi sensor DS18B20, membaca suhu secara periodik, dan mengonversi hasilnya ke dalam satuan derajat Celsius. Nilai suhu yang diperoleh kemudian dikirimkan ke ThingsBoard Cloud melalui protokol komunikasi MQTT (Message Queuing Telemetry Transport) dalam format JSON.

Dengan konfigurasi tersebut, sistem mampu memantau suhu refrigerator secara real-time dan menampilkan datanya secara otomatis di dashboard ThingsBoard. Proses pengiriman data dilakukan secara berulang, memungkinkan pengguna untuk memantau kestabilan suhu penyimpanan obat dari jarak jauh melalui jaringan internet. Kombinasi antara sensor DS18B20, mikrokontroler ESP32-S3, dan bahasa pemrograman Rust menghasilkan sistem pemantauan suhu yang efisien, stabil, serta andal untuk mendukung aplikasi Internet of Things (IoT) di bidang kesehatan dan farmasi

BAB IV

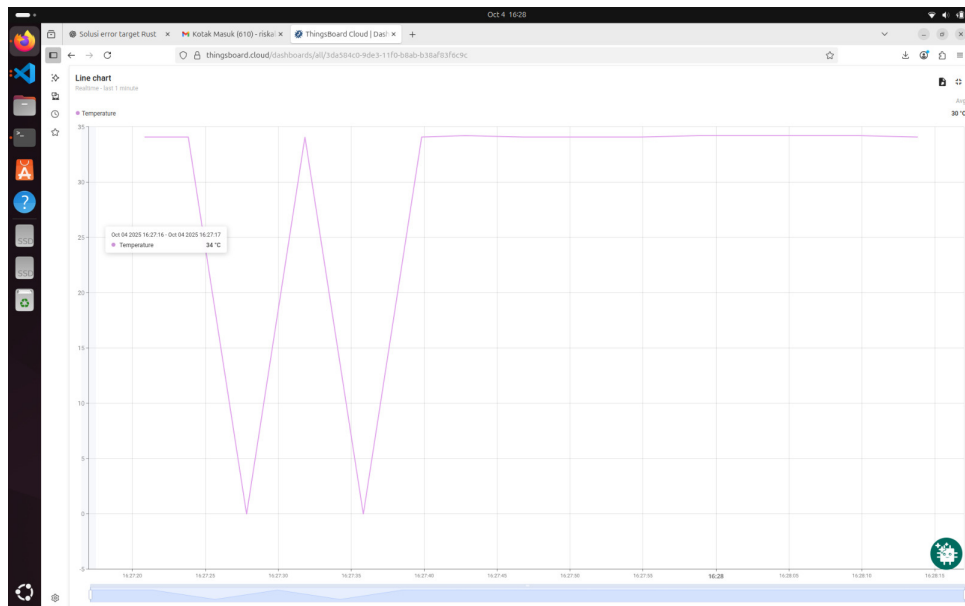
4.1 Hasil Flashing ESP32-S3 dan Pembacaan Sensor DS18B20



Gambar 8: Hasil Flashing ESP32-S3 dan Pembacaan Sensor DS18B20

Gambar 8 merupakan proses flashing pada mikrokontroler ESP32-S3 yang telah berhasil dilakukan. Hal tersebut ditandai dengan munculnya data hasil pembacaan sensor DS18B20 secara periodik di terminal VS Code. Data tersebut menunjukkan nilai temperatur yang dikirim melalui koneksi Wi-Fi ke server ThingsBoard menggunakan protokol MQTT, yang dibuktikan dengan pesan “Data sent to ThingsBoard” dan payload berisi nilai suhu dalam format JSON. Penggunaan ESP32-S3 sebagai perangkat utama sangat tepat karena mikrokontroler ini memiliki prosesor dual-core yang memungkinkan eksekusi tugas paralel secara efisien, serta konektivitas Wi-Fi terintegrasi yang mendukung pengiriman data sensor secara real-time tanpa perlu modul tambahan. Selain itu, implementasi menggunakan bahasa Rust embedded memberikan keunggulan dalam hal *memory safety* dan efisiensi performa, sehingga meminimalkan potensi bug seperti buffer overflow yang umum terjadi pada bahasa pemrograman tingkat rendah lainnya.

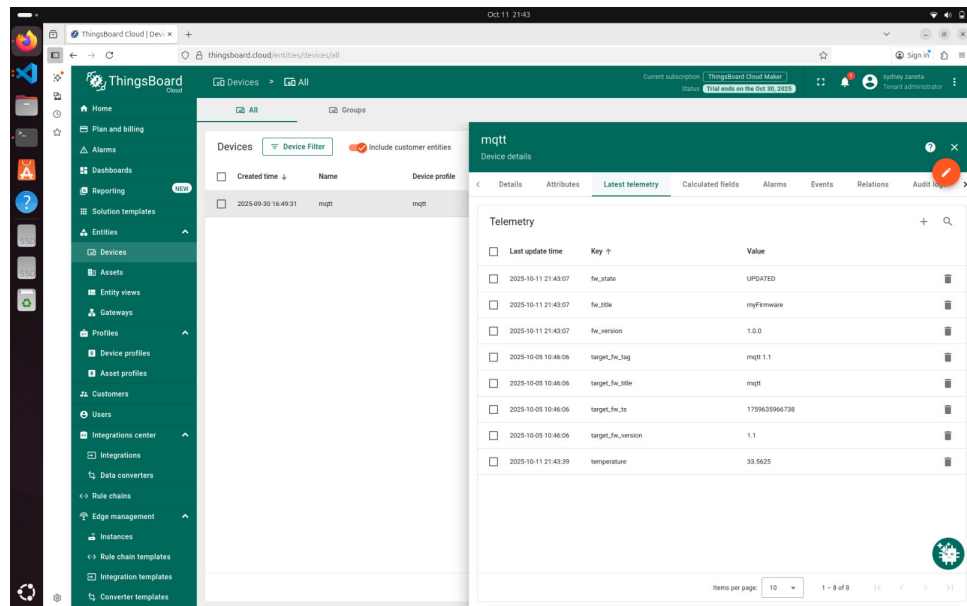
4.2 Tampilan Hasil Pembacaan Sensor DS18B20 pada Thingsboard



Gambar 9: Tampilan Hasil Pembacaan Sensor DS18B20 pada Thingsboard

Berdasarkan hasil grafik pada ThingsBoard Cloud, dapat terlihat bahwa data suhu dari sensor DS18B20 berhasil dikirim dan divisualisasikan secara real-time setelah proses flashing menggunakan VS Code dan pemrograman *Rust embedded* pada ESP32-S3. Grafik tersebut menampilkan fluktuasi nilai suhu dalam satuan derajat celsius ($^{\circ}\text{C}$) yang dibaca oleh sensor dan dikirim melalui koneksi Wi-Fi menggunakan protokol MQTT. Hasil ini menunjukkan bahwa sistem mampu melakukan akuisisi dan transmisi data secara terus-menerus dengan waktu respons yang cepat. Penggunaan ESP32-S3 sebagai perangkat utama memberikan banyak keunggulan, seperti prosesor *dual-core* yang mendukung multitasking antara proses pembacaan sensor dan komunikasi jaringan, serta konektivitas Wi-Fi yang stabil untuk pengiriman data ke *server cloud*. Sementara itu, pemrograman dengan bahasa *Rust embedded* menjamin *memory safety* dan efisiensi performa, sehingga sistem dapat berjalan tanpa gangguan seperti *crash* atau kebocoran memori. Integrasi antara ESP32-S3 dan ThingsBoard Cloud menjadi kombinasi ideal untuk membangun sistem *Internet of Things* (IoT) yang tidak hanya mampu memantau data sensor suhu secara *real-time*, tetapi juga memungkinkan manajemen perangkat dan analisis data secara terpusat melalui antarmuka visual yang interaktif.

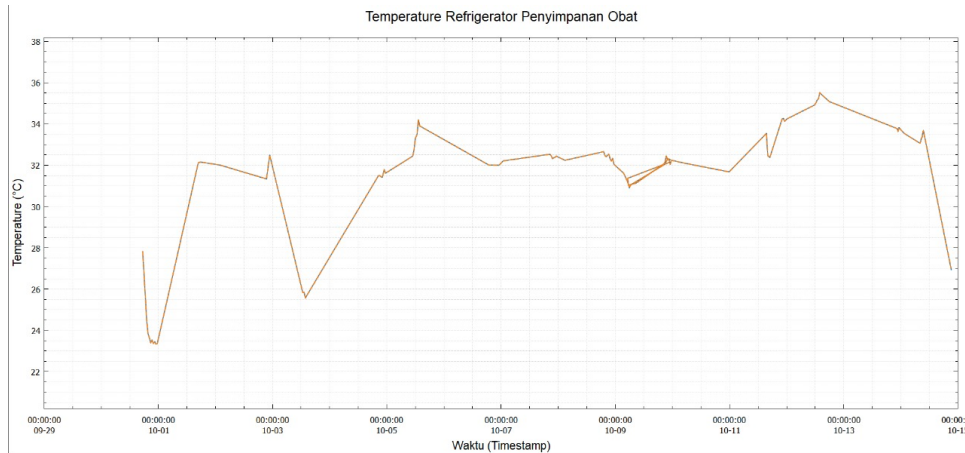
4.3 Over The Air (OTA)



Gambar 10: Hasil Over-The-Air Firmware Deployment

Sistem berhasil melakukan pembaruan firmware jarak jauh (Over-The-Air/OTA) pada perangkat ESP32-S3 melalui platform ThingsBoard Cloud. Pada tampilan tab Latest Telemetry, terlihat bahwa status firmware menunjukkan nilai “UPDATED”, dengan informasi versi firmware (`fw_version = 1.0.0`). serta tag dan judul firmware yang dikonfigurasi melalui MQTT. Hal ini membuktikan bahwa proses OTA telah berjalan dengan baik, di mana pembaruan sistem dilakukan tanpa perlu koneksi fisik langsung ke perangkat. Penerapan OTA ini merupakan fitur penting dalam sistem Internet of Things (IoT) modern karena memberikan kemudahan dalam manajemen dan pemeliharaan perangkat jarak jauh. Penggunaan ESP32-S3 memberikan keuntungan tambahan dengan dukungan prosesor dual-core dan konektivitas Wi-Fi yang handal, sehingga proses pembaruan dapat dilakukan secara efisien dan aman.

4.4 Hasil Gnuplot



Gambar 11: Hasil Gnuplot dari Timestamp pada Real Time Clock (RTC) ESP32-S3 dan Timestamp ThingsBoard

Berdasarkan grafik pembacaan suhu yang ditampilkan, dapat dijelaskan bahwa proses pengambilan data dimulai pada tanggal 30 September 2025. Data yang diperoleh merupakan hasil pengukuran suhu dari sensor DS18B20 yang digunakan untuk memantau suhu pada refrigerator penyimpanan obat. Sensor tersebut dihubungkan dengan mikrokontroler ESP32-S3 yang kemudian mengirimkan hasil pembacaan suhu ke platform ThingsBoard Cloud secara otomatis dan berkala. Rentang waktu pengambilan data berlangsung secara kontinu hingga pertengahan Oktober 2025, menunjukkan bahwa sistem pemantauan bekerja secara aktif dan berkelanjutan sepanjang periode pengujian.

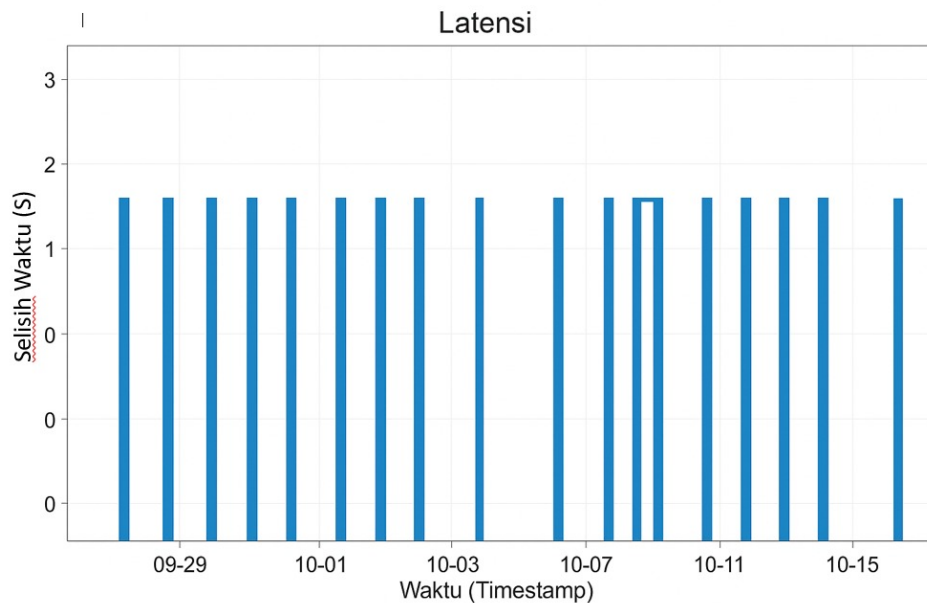
Dari grafik terlihat bahwa nilai suhu yang terbaca mengalami variasi antara sekitar 23°C hingga 36°C, yang menunjukkan adanya fluktuasi suhu yang cukup signifikan selama proses pemantauan. Fluktuasi tersebut sangat mungkin disebabkan oleh perubahan kondisi lingkungan sekitar sensor, seperti intensitas pembukaan pintu lemari pendingin, beban pendingin yang berubah, atau variasi kerja kompresor kulkas. Kondisi ini menggambarkan bahwa pembacaan suhu sangat sensitif terhadap keadaan di sekitar lokasi penempatan sensor, sehingga mencerminkan data yang realistis terhadap situasi sebenarnya.

Selain itu, hasil grafik juga memperlihatkan bahwa sensor DS18B20 terus melakukan pembacaan dan pengiriman data secara konsisten selama periode pengamatan. Tidak terdapat jeda data yang panjang atau kehilangan data yang berarti, menandakan bahwa sistem komunikasi antara sensor, mikrokontroler, dan ThingsBoard Cloud bekerja dengan baik. Hal ini juga mengindikasikan bahwa koneksi jaringan serta konfigurasi over-the-air (OTA) yang diterapkan dalam sistem dapat berjalan stabil tanpa gangguan signifikan selama proses pemantauan berlangsung.

Secara keseluruhan, hasil pengujian ini menunjukkan bahwa proyek berjalan dengan berhasil dan sesuai dengan tujuan yang diharapkan. Sistem mampu melakukan pembacaan suhu secara real-time, mengirimkan data ke cloud, dan menampilkan hasil monitoring dengan

akurat. Keberhasilan ini membuktikan bahwa integrasi antara perangkat keras (sensor dan mikrokontroler) serta perangkat lunak (ThingsBoard Cloud dan sistem OTA) telah berfungsi optimal. Dengan hasil tersebut, sistem ini dinyatakan layak untuk digunakan sebagai solusi pemantauan suhu pada refrigerator penyimpanan obat yang membutuhkan akurasi tinggi dan kestabilan transmisi data secara berkelanjutan.

4.5 Hasil Latency



Gambar 12: Hasil Latency Antara Timestamp pada Real Time Clock (RTC) ESP32-S3 dan Timestamp ThingsBoard

Berdasarkan grafik latensi yang ditunjukkan, dapat dianalisis bahwa data tersebut merepresentasikan selisih waktu antara timestamp sistem dengan waktu aktual yang dibaca dari modul RTC (Real Time Clock). Nilai selisih waktu yang berada pada rentang 0 hingga 2 detik menunjukkan tingkat kestabilan waktu pengiriman data dari sensor DS18B20 menuju platform ThingsBoard Cloud melalui mikrokontroler ESP32-S3. Pada beberapa titik waktu, terlihat adanya selisih sekitar 1 hingga 2 detik. Hal ini dapat diinterpretasikan sebagai adanya gangguan kecil pada proses transmisi data, yang kemungkinan besar disebabkan oleh faktor kestabilan jaringan atau interferensi sinyal ketika mikrokontroler melakukan proses pengiriman ke cloud server.

Meskipun demikian, sebagian besar titik pengukuran menunjukkan nilai selisih waktu yang sangat kecil atau bahkan mendekati nol, yang menandakan bahwa sistem pengiriman data berjalan cukup stabil dan efisien. Kondisi ini menunjukkan bahwa komunikasi antara ESP32-S3 dan ThingsBoard Cloud pada umumnya berjalan dengan baik, di mana keterlambatan pengiriman hanya terjadi sesekali dan tidak bersifat signifikan terhadap performa keseluruhan sistem. Selain itu, kestabilan ini juga menunjukkan bahwa konfigurasi over-the-air (OTA) update serta

proses pembacaan sensor DS18B20 tidak menimbulkan latensi tambahan yang berarti dalam sistem.

Dari hasil pengamatan terhadap tren latensi tersebut, dapat disimpulkan bahwa variasi waktu yang terjadi cenderung bersifat acak dan tidak membentuk pola tertentu, sehingga kemungkinan besar dipengaruhi oleh faktor eksternal seperti fluktuasi kekuatan sinyal Wi-Fi, beban jaringan pada saat pengiriman data, atau kondisi lingkungan yang memengaruhi kualitas koneksi. Hal ini mengindikasikan bahwa sistem sudah cukup tangguh dalam menghadapi perubahan kondisi jaringan tanpa mengakibatkan kesalahan atau kehilangan data yang signifikan. Dengan demikian, sistem mampu menjaga kontinuitas komunikasi data meskipun terdapat gangguan sementara.

Secara keseluruhan, sistem yang dikembangkan telah menunjukkan performa yang baik dalam aspek keandalan, kestabilan waktu, serta keakuratan pengiriman data. Nilai latensi yang rendah memperlihatkan bahwa arsitektur komunikasi antara sensor, mikrokontroler, dan ThingsBoard Cloud telah diimplementasikan dengan efektif. Hasil ini menjadi bukti bahwa integrasi perangkat keras dan perangkat lunak, termasuk fitur OTA, berjalan dengan baik sehingga sistem dapat diandalkan untuk aplikasi pemantauan suhu secara real-time dan berskala besar di masa mendatang.

BAB V

SARAN DAN KESIMPULAN

5.1 Kesimpulan

Proyek ini telah berhasil mengimplementasikan sistem berbasis ESP32-S3 dengan dukungan pemrograman Rust embedded. Proyek ini menunjukkan bahwa ESP32-S3 mampu menjalankan fungsinya sebagai perangkat utama yang efisien dan andal, dengan keunggulan pada prosesor dual-core, serta konektivitas Wi-Fi yang stabil untuk mengirimkan data sensor DS18B20 secara real-time ke platform ThingsBoard Cloud. Melalui integrasi tersebut, sistem tidak hanya berhasil menampilkan data suhu secara langsung pada dashboard ThingsBoard, tetapi juga telah mampu mendukung proses manajemen perangkat dan pembaruan firmware jarak jauh (OTA) yang terbukti berjalan sukses hingga status “UPDATED”. Selain itu, sistem juga mampu beroperasi secara mandiri menggunakan daya adaptor, menunjukkan kestabilan kerja perangkat di luar koneksi komputer. Keberhasilan pembacaan timestamp dari Real Time Clock (RTC) dan timestamp ThingsBoard yang kemudian divisualisasikan menggunakan Gnuplot, menegaskan bahwa sistem mampu menghasilkan data yang akurat untuk analisis latency jaringan dan sinkronisasi waktu. Dengan demikian, proyek ini dapat membuktikan implementasi nyata dari konsep IoT modern yang meliputi akuisisi data, komunikasi real-time, manajemen perangkat, pembaruan firmware jarak jauh, serta analisis performa sistem berbasis waktu nyata.

5.2 Saran

Berdasarkan hasil pengembangan yang telah dilakukan, terdapat beberapa saran untuk pengembangan lebih lanjut agar sistem menjadi lebih robust dan fungsional:

1. **Notifikasi dan Alarm Otomatis:** Integrasikan fitur alert system pada ThingsBoard (misalnya melalui email, Telegram bot, atau notifikasi web) yang aktif ketika suhu melewati ambang batas tertentu. Ini akan meningkatkan nilai praktis sistem untuk aplikasi kesehatan atau industri farmasi.
1. **Keamanan Data dan Autentikasi:** Untuk memperkuat sistem, gunakan koneksi MQTT over TLS/SSL dan tambahkan autentikasi dua arah (device token dan certificate). Hal ini penting untuk mencegah manipulasi atau akses tidak sah pada jaringan IoT.
1. **Optimasi Penggunaan Daya dan Mode Sleep:** Untuk efisiensi energi, terutama jika sistem digunakan di lapangan, manfaatkan deep sleep mode pada ESP32-S3. Dengan pengaturan wake-up periodik berdasarkan RTC, perangkat bisa menghemat daya tanpa mengorbankan kontinuitas data.

DAFTAR PUSTAKA

- Ammar Haziq Annas, A. A. (2024). *Cloud-Based IoT System for Real-Time Harmful Algal Bloom Monitoring: Seamless ThingsBoard Integration via MQTT and REST API*. IEEE 22nd Student Conference on Research and Development (SCORED).
- Bangyan Huang, J. L. (2012). *The Reading Data Error Analysis of 1-wire Bus Digital Temperature Sensor DS18B20*. Proceedings of 2012 International Conference.
- Elham Okhovat, M. B. (2021). *Monitoring the Smart City Sensor Data Using Thingsboard and Node-Red*. IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/IOP/SCI).
- Hongxia XU, W. W. (2023). *Design of Portable Refrigerator based on DS18B20 Temperature Sensor*. ISSTC.
- J Bino, M. M. (2024). *FireEye: An IoT-Based Fire Alarm and Detection System for Enhanced Safety*. Intelligent Systems and Machine Learning Conference (ISML).
- Kadek Darmaastawan, I. W. (2024). *Towards User Friendly Smart Precision Farming: Assessing ThingsBoard as an Interface for IoT Based Farming System Using System Usability Scale*. ICSCC.
- Litayem, N. (2024). *Scalable Smart Home Management with ESP32-S3: A Low-Cost Solution for Accessible Home Automation*. International Conference on Computer and Applications (ICCA).
- Luqiang Sun, H. Z. (2023). *The Research on Security Technology of Earthquake Warning Information Release Based on Zero Trust MQTT Protocol*. International Conference on Mobile Internet, Cloud Computing and Information Security (MICCIS).
- Manas Mishra, A. S. (2024). *ESP32-S3 DevKit-C Integration for Enhanced Road Safety for Pothole Detection and Nighttime Driving Assistance*. 2nd International Conference on Disruptive Technologies (ICDT).
- Matthew Henschke, X. W. (2020). *Data Visualization for Wireless Sensor Networks Using ThingsBoard*. Wireless and Optical Communication Conference (WOCC).
- Meijun Qu, K. Z. (2022). *Research on Accurate OTA Testing for Large-Form-Factor 5G IoT Devices in Medium-Sized Anechoic Chamber*. IEEE Antennas and Wireless Propagation Letters.
- Ranu Wijaya, A. R. (2025). *A Novel LoRa-Based Wireless Communication Module for*

Industrial Data Acquisition-Telemetry. IEEE.

Samy H. Sharf, R. A. (2021). *An Efficient OTA Firmware Updating Architecture Based on LoRa Suitable for Agricultural IoT Applications. 2021 International Conference on Microelectronics (ICM).*

Syaiful Andy, B. R. (2017). *Attack Scenarios and Security Analysis of MQTT Communication Protocol in IoT System. Proceedings of EECSI.*

Thomas Pfitzinger, H. W. (2023). *Embedded Real-Time Human Activity Recognition on an ESP32-S3 Microcontroller Using Ambient Audio Data. The 12th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications.*

Weiping Wang, Y. Z. (2024). *MQTT Protocol and Implementation of Equipment Management System for Industrial Internet of Things. Proceedings of the 43rd Chinese Control Conference.*

A.S, S. (2017). *Intelligent Refrigerator Using Artificial Intelligence. International Conference on Intelligent Systems and Control (ISCO).*

Ali Hammadeh, A. Z.-L. (2021). *Time Series Ordinal Regression for Supporting the Storage of Temperature Sensitive Medication in Domestic Refrigerators. International Conference on Data Mining Workshops (ICDMW).*

Ammar Haziq Annas, A. A. (2024). *Cloud-Based IoT System for Real-Time Harmful Algal Bloom Monitoring: Seamless ThingsBoard Integration via MQTT and REST API. IEEE 22nd Student Conference on Research and Development (SCORED).*

Thomas Pfitzinger, H. W. (2023). *Embedded Real-Time Human Activity Recognition on an ESP32-S3 Microcontroller Using Ambient Audio Data. The 12th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications.*

LAMPIRAN

```
use anyhow::Result;
use esp_idf_hal::delay::Delay;
use esp_idf_hal::gpio::*;
use esp_idf_hal::prelude::*;
use esp_idf_svc::log::EspLogger;
use esp_idf_svc::eventloop::EspSystemEventLoop;
use esp_idf_svc::nvs::EspDefaultNvsPartition;
use esp_idf_svc::wifi::{AuthMethod, BlockingWifi, ClientConfiguration,
    Configuration, EspWifi};
use log::*;
use onewire::{Device, DeviceSearch, OneWire};
use serde_json::json;
use alloc::string::ToString;
use alloc::ffi::CString;

extern crate alloc;

// Structure to represent temperature in Celsius
#[derive(Debug)]
struct Celsius(f32);

impl Celsius {
    pub fn as_f32(&self) -> f32 {
        self.0
    }
}

// Convert milliseconds to FreeRTOS ticks
#[inline(always)]
fn ms_to_ticks(ms: u32) -> u32 {
    (ms as u64 * esp_idf_sys::configTICK_RATE_HZ as u64 / 1000) as u32
}

// MQTT client wrapper
struct SimpleMqttClient {
    client: *mut esp_idf_sys::esp_mqtt_client,
}

impl SimpleMqttClient {
    fn new(broker_url: &str, username: &str, password: &str, client_id: &
        str) -> Result<Self> {
        unsafe {
            let broker_url_cstr = CString::new(broker_url)?;
            let username_cstr = CString::new(username)?;
            let password_cstr = CString::new(password)?;
```

```

    let client_id_cstr = CString::new(client_id)?;

    let config = esp_idf_sys::esp_mqtt_client_config_t {
        broker: esp_idf_sys::esp_mqtt_client_config_t_broker_t {
            address: esp_idf_sys::
esp_mqtt_client_config_t_broker_t_address_t {
                uri: broker_url_cstr.as_ptr() as *const u8,
                ..core::mem::zeroed()
            },
            ..core::mem::zeroed()
        },
        credentials: esp_idf_sys::
esp_mqtt_client_config_t_credentials_t {
            username: username_cstr.as_ptr() as *const u8,
            client_id: client_id_cstr.as_ptr() as *const u8,
            authentication: esp_idf_sys::
esp_mqtt_client_config_t_credentials_t_authentication_t {
                password: password_cstr.as_ptr() as *const u8,
                ..core::mem::zeroed()
            },
            ..core::mem::zeroed()
        },
        ..core::mem::zeroed()
    };

    let client = esp_idf_sys::esp_mqtt_client_init(&config);
    if client.is_null() {
        return Err(anyhow::anyhow!("Failed to initialize MQTT
client"));
    }

    let err = esp_idf_sys::esp_mqtt_client_start(client);
    if err != esp_idf_sys::ESP_OK {
        return Err(anyhow::anyhow!("Failed to start MQTT client,
error code: {}", err));
    }

    esp_idf_sys::vTaskDelay(ms_to_ticks(5000));
    Ok(Self { client })
}

pub fn publish(&self, topic: &str, payload: &str) -> Result<()> {
    unsafe {
        let topic_cstr = CString::new(topic)?;
        let payload_cstr = CString::new(payload)?;
        let msg_id = esp_idf_sys::esp_mqtt_client_publish(

```

```

        self.client,
        topic_cstr.as_ptr(),
        payload_cstr.as_ptr(),
        0,
        1,
        false
    );
    if msg_id < 0 {
        return Err(anyhow::anyhow!("Failed to publish message"));
    }
    Ok(())
}
}

fn main() -> Result<()> {
    EspLogger::initialize_default();

    let peripherals = Peripherals::take().unwrap();
    let sysloop = EspSystemEventLoop::take()?;
    let nvs = EspDefaultNvsPartition::take()?;

    // WiFi setup
    let mut wifi = BlockingWifi::wrap(EspWifi::new(peripherals.modem,
        sysloop.clone(), Some(nvs))?, sysloop)?;
    wifi.set_configuration(&Configuration::Client(ClientConfiguration {
        ssid: "MyWiFi".try_into().unwrap(),
        password: "MyPassword".try_into().unwrap(),
        auth_method: AuthMethod::WPA2Personal,
        ..Default::default()
    }))?;
    wifi.start()?;
    wifi.connect()?;
    wifi.wait_netif_up()?;

    info!("Connected to Wi-Fi");

    // MQTT setup
    let mqtt = SimpleMqttClient::new("mqtt://demo.thingsboard.io", "user",
        "pass", "esp32-client"?;

    let mut delay = Delay::new_default();

    loop {
        let temp_celsius = Celsius(25.3);
        let payload = json!({
            "temperature": temp_celsius.as_f32(),

```

```
        "timestamp": 1700000000
    });

    mqtt.publish("v1/devices/me/telemetry", &payload.to_string())?;
    info!("Data published: {:?}", payload);

    delay.delay_ms(5000u32);
}
}
```

Listing 1: Kode Program Rust ESP32-S3 untuk IoT