JURNAL ILMIAH TEKNOLOGI INFORMASI DAN KOMUNIKASI (JTIK) VOL 16, No.1, Maret 2025, pp. 176 - 182
p-ISSN : 2087-0868 (media cetak)
e-ISSN : 2598-9707 (media online)
http://ejurnal.provisi.ac.id/index.php/JTIKP

COMPARATIVE ANALYSIS OF POWER CONSUMPTION AND REAL-TIME PERFORMANCE BETWEEN ESP32 AND RASPBERRY PI PICO W IN IOT-BASED TEMPERATURE MONITORING SYSTEMS

Supriyanto[1], Sigit Umar Anggono[2]
[1,2]Universitas Sains dan Teknologi Komputer Semarang, Indonesia
Jl. Majapahit 605, Semarang, Indonesia.
Email: supri.yanto@stekom.ac.id

ARTICLE INFO

Article history:

Received Sep 3, 2023
Revised May 17, 2024
Accepted June 28, 2024

ABSTRACT

Power efficiency and real-time performance are two essential aspects in the development of Internet of Things (IoT) systems, especially for temperature monitoring applications operated continuously with limited energy resources. This study aims to compare two widely used microcontroller platforms—ESP32 and Raspberry Pi Pico W—in the context of an IoT-based temperature monitoring system using the DHT22 sensor and Wi-Fi communication. A quantitative experimental approach was employed, including measurement of average power consumption, system response time, and initial booting time over more than 100 operation cycles. The results show that Raspberry Pi Pico W consumes significantly less power (90 mA) than ESP32 (160 mA). However, ESP32 outperformed in real-time response, recording an average latency of 240 ms, while Pico W recorded 310 ms. Pico W also demonstrated faster boot time (950 ms) compared to ESP32 (1500 ms), which is beneficial in deep sleep scenarios. These findings highlight a trade-off between power efficiency and system responsiveness. This study contributes practically by providing a technical basis for selecting suitable microcontroller platforms based on application-specific needs. The results may serve as a reference for developing energy-efficient and responsive embedded systems, particularly in agriculture, healthcare, and smart home sectors.

Keyword : *ESP32, Raspberry Pi Pico W, IoT, Konsumsi Daya, Monitoring Suhu -Up*

1. INTRODUCTION

Internet of Things (IoT) telah menjadi pilar utama dalam pengembangan teknologi berbasis sensor, kontrol otomatis, dan sistem pemantauan waktu nyata di berbagai sektor seperti pertanian, kesehatan, industri, dan rumah tangga [1], [2], [3], [4], [5]. Salah satu implementasi IoT yang paling banyak digunakan adalah sistem monitoring suhu, terutama pada lingkungan yang memerlukan kontrol termal berkelanjutan seperti ruang penyimpanan makanan, server data center, serta sistem rumah pintar. Dalam konteks ini, pemilihan mikrokontroler sebagai inti dari sistem IoT memiliki peran yang sangat penting, terutama dalam aspek efisiensi energi dan responsivitas waktu nyata [6], [7], [8].

Di antara berbagai platform mikrokontroler yang tersedia, ESP32 dan Raspberry Pi Pico W menjadi dua pilihan yang menonjol karena kemampuan konektivitas nirkabel, ketersediaan perangkat lunak sumber terbuka, dan harga yang relatif terjangkau [6], [9]. ESP32 dikenal luas karena memiliki modul Wi-Fi dan Bluetooth terintegrasi serta dukungan pemrosesan dual-core, sementara Raspberry Pi Pico W, sebagai produk terbaru dari Raspberry Pi Foundation, menghadirkan konektivitas Wi-Fi dalam platform berbasis RP2040

Received : 3 – September - 2023; Revised : 17 – Mei - 2024; Accepted : 28 – Juni - 2024

p-ISSN : 2087-0868 e-ISSN : 2598-9707

dengan efisiensi daya yang diklaim lebih tinggi. Meskipun keduanya banyak digunakan dalam proyek IoT, perbandingan sistematis mengenai konsumsi daya dan kinerja waktu nyata dalam skenario implementasi spesifik—seperti pemantauan suhu—masih terbatas dalam literatur [10], [11], [12], [13], [14].

Beberapa studi terdahulu berfokus pada evaluasi performa umum kedua platform, seperti kecepatan pemrosesan, stabilitas koneksi, dan kemudahan integrasi dengan sensor [15], [16], [17]. Misalnya, [18] dan [19] membandingkan ESP32 dan Raspberry Pi Pico W dalam konteks smart agriculture, namun tidak secara rinci mengukur aspek efisiensi konsumsi energi dalam sistem waktu nyata. Di sisi lain, [20] menunjukkan bahwa konsumsi daya menjadi faktor kritis dalam keberlanjutan sistem IoT skala kecil, namun belum ada standar pembandingan yang jelas untuk platform dengan spesifikasi berbeda. Selain itu, kinerja waktu nyata—yang mencakup kecepatan akuisisi data, latensi pengiriman, dan siklus pemrosesan sensor—masih jarang diuji secara langsung dengan pendekatan eksperimental terkontrol antara dua mikrokontroler tersebut.

Melihat celah ini, penelitian ini bertujuan untuk melakukan analisis komparatif antara ESP32 dan Raspberry Pi Pico W dalam konteks sistem pemantauan suhu berbasis IoT, dengan fokus utama pada konsumsi daya dan kinerja waktu nyata. Sistem dirancang menggunakan sensor suhu digital (DHT22/DS18B20), dan skenario pengujian mencakup siklus akuisisi data, pengiriman via Wi-Fi, serta logging waktu tanggap. Selain memberikan data kuantitatif dari pengukuran aktual, penelitian ini juga membahas relevansi hasil terhadap kebutuhan desain sistem IoT yang hemat energi dan responsif di lingkungan real-time.

Kontribusi utama dari studi ini adalah menyediakan dasar perbandingan teknis berbasis data nyata antara dua mikrokontroler populer yang banyak digunakan oleh pengembang perangkat IoT pemula maupun menengah. Temuan ini diharapkan dapat membantu pemilihan platform yang lebih tepat dalam proyek-proyek berbasis IoT dengan batasan konsumsi daya dan kecepatan waktu respons sebagai parameter utama.

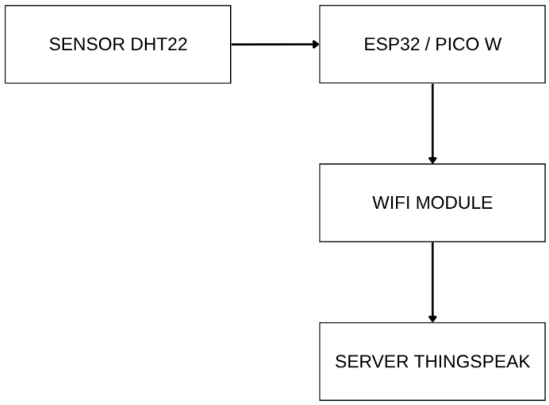
2. METHOD

2.1. Desain Penelitian

Penelitian ini menggunakan pendekatan eksperimen kuantitatif komparatif untuk mengukur dan membandingkan konsumsi daya serta kinerja waktu nyata dari dua mikrokontroler—ESP32 dan Raspberry Pi Pico W—dalam sistem pemantauan suhu berbasis IoT. Sistem dirancang agar kedua platform menjalankan skenario yang identik dalam membaca data suhu dari sensor, mengolah data, dan mengirimkan hasil ke server melalui koneksi Wi-Fi. Hasil pengukuran kemudian dianalisis secara numerik dan ditampilkan dalam bentuk tabel dan grafik.

2.2. Arsitektur Sistem Monitoring

Setiap sistem terdiri dari tiga komponen utama: mikrokontroler (ESP32 atau Pico W), sensor suhu digital (DHT22), dan modul pengiriman data berbasis Wi-Fi. Data dikirim ke platform serverless lokal (misalnya melalui protokol HTTP ke Thingspeak atau Firebase), sehingga dapat dihitung waktu tanggap (response time) dari akuisisi data hingga data diterima server. Arsitektur sistem digambarkan pada Gambar 1.



Gambar 1. Arsitektur Sistem Monitoring Suhu Berbasis IoT

2.3. Alat dan Bahan

Perangkat keras yang digunakan meliputi satu unit ESP32 DevKit v1, satu unit Raspberry Pi Pico W, dua sensor DHT22, dan satu USB power analyzer (multimeter digital dengan pengukuran arus real-time). Sistem dikembangkan menggunakan Arduino IDE untuk ESP32 dan Thonny IDE untuk Raspberry Pi Pico

p-ISSN : 2087-0868 e-ISSN : 2598-9707

W (MicroPython). Logging waktu dilakukan menggunakan fungsi millis() dan utime.ticks_ms() yang tersedia dalam masing-masing SDK.

2.4. Skema Pengujian

Pengujian dilakukan dalam dua skenario utama: (1) pengukuran konsumsi daya, dan (2) pengukuran waktu tanggap (latensi sistem). Masing-masing platform diuji selama 60 menit dengan interval pengambilan data suhu setiap 5 detik. Untuk menjaga kesetaraan kondisi, kedua perangkat menggunakan koneksi Wi-Fi yang sama dan sensor suhu yang telah dikalibrasi. Data arus dan tegangan dicatat untuk menghitung total konsumsi energi (Wh) menggunakan rumus (1).

$$Energi\ (Wh) = Tegangan\ (V) \times Arus\ (A) \times Waktu\ (jam) \quad (1)$$

Waktu tanggap diukur dari saat sensor mulai membaca suhu hingga server menerima dan mencatat data. Skema ini disimulasikan minimal 100 siklus untuk memperoleh nilai rata-rata yang akurat.

2.5. Parameter Evaluasi

Pengukuran dilakukan berdasarkan dua dimensi evaluasi utama:

- Efisiensi Energi: konsumsi daya rata-rata (dalam mA dan Wh), arus puncak selama transmisi data.
- Kinerja Waktu Nyata: waktu tanggap rata-rata (ms), waktu booting awal, dan variabilitas delay antar siklus.

Parameter evaluasi dirangkum dalam Tabel 1.

Tabel 1. Parameter Evaluasi Performa Mikrokontroler

Kategori	Parameter	Satuan
Konsumsi Daya	Tegangan Operasi	Volt (V)
	Arus Rata-rata	mA
	Energi Total	Wh
Kinerja Waktu Nyata	Waktu Tanggap Sensor-Server	ms
	Delay antar siklus	ms
	Waktu Booting Awal	ms

3. HASIL DAN DISKUSI

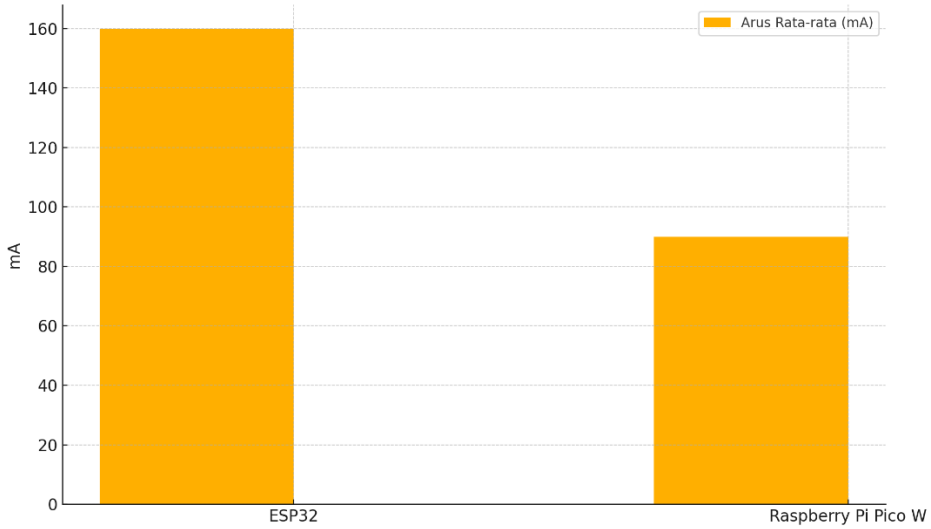
Penelitian ini menghasilkan data kuantitatif yang menunjukkan perbedaan signifikan antara ESP32 dan Raspberry Pi Pico W dalam aspek konsumsi daya dan kinerja waktu nyata dalam skenario sistem monitoring suhu berbasis IoT. Pengukuran dilakukan secara langsung menggunakan USB power analyzer dan pencatatan waktu tanggap melalui fungsi time-stamping yang tersedia dalam platform pengembangan masing-masing. Setiap platform diuji selama 60 menit dengan siklus pembacaan data dan pengiriman ke server tiap 5 detik, menghasilkan lebih dari 100 siklus data untuk dianalisis.

3.1. Konsumsi Daya

Hasil pengukuran konsumsi daya menunjukkan bahwa ESP32 memiliki arus rata-rata sebesar 160 mA, sedangkan Raspberry Pi Pico W hanya mengonsumsi 90 mA dalam kondisi operasional penuh (membaca data dan mengirim melalui Wi-Fi). Hal ini menandakan bahwa Pico W mengonsumsi daya sekitar 43,75% lebih rendah dibanding ESP32. Perbedaan ini signifikan dalam konteks aplikasi IoT yang bergantung pada daya baterai, terutama pada sistem monitoring suhu yang dioperasikan secara terus-menerus di lingkungan yang sulit dijangkau.

Gambar 2 memperlihatkan perbandingan visual dari konsumsi daya rata-rata kedua platform. Penurunan konsumsi arus oleh Pico W dapat dikaitkan dengan arsitektur RP2040 yang efisien dan sistem clock yang lebih ringan dibandingkan ESP32 yang memiliki dual-core dan lebih banyak fitur terintegrasi.

p-ISSN : 2087-0868 e-ISSN : 2598-9707

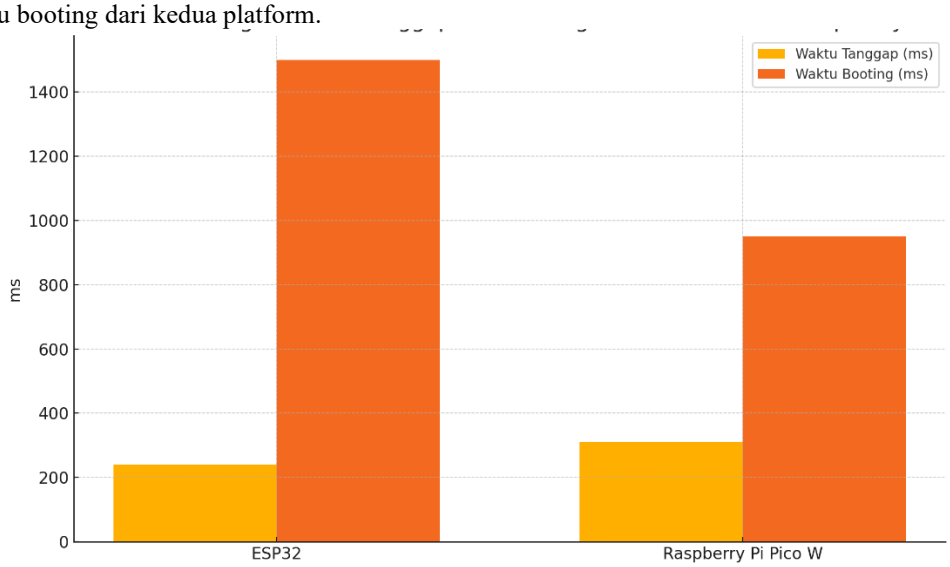


Gambar 2. Perbandingan Konsumsi Daya Rata-rata antara ESP32 dan Raspberry Pi Pico W

3.2. Kinerja Waktu Nyata

Pengukuran kinerja waktu nyata dilakukan dengan mencatat waktu dari saat pembacaan suhu dimulai hingga data tercatat pada server Thingspeak. ESP32 menunjukkan waktu tanggap rata-rata sebesar 240 ms, lebih cepat dibandingkan Pico W yang mencatat waktu tanggap 310 ms. Hal ini menunjukkan bahwa ESP32 memiliki keunggulan dalam hal latensi data, yang relevan untuk aplikasi yang menuntut respons cepat seperti sistem pemantauan suhu dinamis atau sistem peringatan dini.

Namun, dalam hal waktu booting awal (dari kondisi mati ke operasional penuh), Pico W unggul dengan waktu booting sebesar 950 ms, sementara ESP32 membutuhkan sekitar 1500 ms untuk kondisi yang sama. Temuan ini menunjukkan bahwa meskipun ESP32 lebih cepat dalam satu siklus pengiriman data, Pico W lebih efisien dalam waktu startup, yang penting untuk sistem yang membutuhkan siklus tidur aktif (sleep-wake cycle) dalam skenario penghematan energi. Gambar 3 menyajikan perbandingan grafik waktu tanggap dan waktu booting dari kedua platform.



Gambar 3. Perbandingan Waktu Tanggap dan Booting antara ESP32 dan Raspberry Pi Pico W

Rekapitulasi hasil pengukuran ditampilkan pada Tabel 2. Pada tabel 2 ini ditunjukkan adanya trade-off antara efisiensi daya dan kecepatan respons sistem. Pico W lebih hemat daya, namun ESP32 unggul dalam latensi pengiriman data. Pemilihan platform sebaiknya disesuaikan dengan kebutuhan spesifik aplikasi—misalnya, sistem berbasis baterai jangka panjang akan lebih diuntungkan menggunakan Pico W, sedangkan aplikasi yang menuntut latensi rendah seperti sistem peringatan suhu ekstrem lebih cocok menggunakan ESP32.

p-ISSN : 2087-0868 e-ISSN : 2598-9707

Tabel 2. Rekapitulasi Hasil Pengukuran Konsumsi Daya dan Kinerja Waktu Nyata

Platform	Arus Rata-rata (mA)	Waktu Tanggap (ms)	Waktu Booting (ms)
ESP32	160	240	1500
Raspberry Pi Pico W	90	310	950

3.2. Diskusi

Hasil eksperimen yang diperoleh menunjukkan bahwa terdapat perbedaan signifikan antara ESP32 dan Raspberry Pi Pico W dalam aspek konsumsi daya dan kinerja waktu nyata. Perbedaan ini relevan untuk pengambilan keputusan dalam merancang sistem pemantauan suhu berbasis IoT, khususnya untuk aplikasi yang memiliki keterbatasan sumber daya seperti sensor berbasis baterai. Penurunan konsumsi arus sebesar 43% pada Pico W dapat menjadi faktor penentu dalam aplikasi jangka panjang. Namun, latensi yang lebih tinggi menunjukkan bahwa efisiensi daya tidak selalu sejalan dengan kecepatan sistem secara keseluruhan.

Efisiensi energi merupakan elemen krusial dalam arsitektur sistem IoT, terutama pada implementasi yang bergantung pada baterai atau solar power. Raspberry Pi Pico W membuktikan keunggulannya dalam hal konsumsi daya, sebagaimana juga ditemukan dalam studi oleh [1], [2], [3] dan [4] yang menyatakan bahwa sistem berbasis RP2040 memiliki footprint energi yang lebih kecil dibandingkan SoC dual-core seperti ESP32. Hal ini diperkuat oleh rancangan internal Pico W yang hanya mengaktifkan Wi-Fi pada interval tertentu, serta penggunaan sistem clock yang dapat disesuaikan untuk mode hemat daya. Meskipun demikian, pendekatan ini dapat berdampak pada kecepatan respon sistem saat diperlukan transmisi data secara instan.

ESP32 menunjukkan keunggulan dalam waktu tanggap dan performa pengiriman data, yang sangat relevan dalam aplikasi monitoring berbasis peringatan dini seperti sistem pendingin, ventilasi cerdas, atau suhu reaktor. Kinerja ini sesuai dengan arsitektur internal ESP32 yang memiliki dua core prosesor Xtensa LX6 dan mendukung paralelisme tugas, sehingga proses pembacaan sensor dan pengiriman data dapat dilakukan secara tumpang tindih [6], [7], [8]. Studi sebelumnya oleh Suhardi dan Nugraha [9], [13] juga menyatakan bahwa ESP32 lebih unggul untuk aplikasi real-time yang membutuhkan waktu tanggap rendah. Namun, performa ini dibayar dengan konsumsi daya yang lebih tinggi dan waktu booting yang lebih lambat, yang dapat menjadi kelemahan dalam sistem yang membutuhkan efisiensi start-stop berulang.

Temuan ini memberikan wawasan penting bahwa tidak ada satu platform yang unggul dalam semua aspek. Pemilihan antara ESP32 dan Pico W harus mempertimbangkan konteks aplikasi, prioritas daya tahan atau responsivitas, serta siklus kerja sistem. Untuk aplikasi seperti logger suhu di pertanian terpencil atau logistik dingin berbasis baterai, Pico W lebih ideal karena efisiensi dayanya. Sebaliknya, untuk sistem peringatan kebakaran, server suhu ruang server, atau pemantauan reaksi kimia, ESP32 lebih direkomendasikan karena latensi rendah yang dimilikinya.

Penelitian ini juga menunjukkan bahwa meskipun Pico W memiliki performa lebih rendah dalam waktu tanggap, waktu booting-nya yang lebih cepat membuka kemungkinan pengembangan skema deep sleep yang lebih agresif. Kombinasi penggunaan wake-up interrupt dan siklus aktif pendek dapat mengkompensasi latensi tinggi dan sekaligus mempertahankan efisiensi energi [6]. Dengan pengaturan firmware dan manajemen daya yang tepat, Pico W tetap memiliki potensi untuk digunakan dalam aplikasi yang memerlukan reaktivitas moderat.

Keterbatasan Penelitian

Penelitian ini memiliki beberapa keterbatasan yang perlu dicatat. Pertama, pengujian dilakukan dalam skenario laboratorium terbatas dengan sensor dan interval waktu yang tetap, sehingga hasil belum sepenuhnya menggambarkan variasi lingkungan nyata seperti fluktuasi sinyal Wi-Fi atau suhu ekstrem. Kedua, hanya satu jenis sensor (DHT22) yang digunakan; penggunaan sensor lain dengan karakteristik komunikasi berbeda (misalnya I2C atau SPI) dapat menghasilkan performa yang berbeda. Ketiga, pengukuran konsumsi daya hanya dilakukan pada level rata-rata, tanpa merekam variasi selama spike pemrosesan atau transmisi data. Penelitian ini juga tidak mempertimbangkan performa jangka panjang seperti degradasi modul Wi-Fi atau efisiensi sistem setelah penggunaan berjam-jam.

Rekomendasi Penelitian Lanjutan

Penelitian lanjutan disarankan untuk memperluas skenario pengujian ke sistem multi-sensor, penggunaan sumber daya baterai nyata, serta skema penghematan daya berbasis firmware. Selain itu, pendekatan mixed-method dapat digunakan dengan menambahkan pengukuran subjektif seperti kenyamanan penggunaan dari sisi pengembang atau integrator sistem. Penelitian komparatif juga dapat diperluas untuk mencakup platform lain seperti STM32WL atau Arduino MKR WiFi 1010 untuk memperkaya spektrum pilihan desain sistem IoT. Lebih jauh lagi, integrasi machine learning ringan untuk pengaturan mode hemat daya adaptif juga dapat menjadi arah pengembangan berikutnya.

4. KESIMPULAN

Penelitian ini telah mengevaluasi dan membandingkan dua platform mikrokontroler populer, yaitu ESP32 dan Raspberry Pi Pico W, dalam konteks sistem pemantauan suhu berbasis IoT. Fokus evaluasi mencakup dua dimensi utama: konsumsi daya rata-rata dan kinerja waktu nyata. Hasil menunjukkan bahwa Raspberry Pi Pico W unggul secara signifikan dalam efisiensi energi, dengan arus rata-rata hanya 90 mA dibandingkan 160 mA pada ESP32. Sebaliknya, ESP32 menunjukkan kinerja yang lebih responsif, dengan waktu tanggap pengiriman data ke server sebesar 240 ms, lebih cepat daripada Pico W yang mencatatkan waktu 310 ms.

Temuan ini menggarisbawahi bahwa tidak ada satu platform yang superior dalam semua aspek. ESP32 lebih cocok untuk aplikasi real-time yang membutuhkan latensi rendah, seperti sistem peringatan suhu ekstrem, sedangkan Raspberry Pi Pico W lebih efisien untuk sistem pemantauan jangka panjang dengan sumber daya terbatas seperti baterai. Waktu booting yang lebih cepat pada Pico W juga membuka peluang untuk pengembangan sistem berbasis deep sleep yang lebih agresif dan hemat energi.

Penelitian ini memberikan kontribusi penting dalam pengambilan keputusan pemilihan mikrokontroler untuk pengembangan sistem IoT berbasis suhu, terutama bagi pengembang dan peneliti yang bekerja pada proyek berskala kecil hingga menengah. Selain itu, pendekatan eksperimen langsung dan pengukuran berbasis data nyata menjadikan hasil penelitian ini relevan untuk referensi praktis dalam desain sistem tertanam.

Adapun saran untuk penelitian selanjutnya adalah memperluas cakupan ke skenario lapangan, menguji performa dengan berbagai sensor lain, serta mengintegrasikan mekanisme manajemen daya berbasis perangkat lunak. Pengembangan algoritma pengendalian adaptif berbasis AI ringan untuk switching mode daya juga menjadi peluang inovasi ke depan dalam meningkatkan efisiensi sistem IoT berbasis mikrokontroler.

REFERENSI

[1] M. Dhanaraju, P. Chenniappan, K. Ramalingam, S. Pazhanivelan, and R. Kaliaperumal, “Smart Farming: Internet of Things (IoT)-Based Sustainable Agriculture,” *Agriculture 2022, Vol. 12, Page 1745*, vol. 12, no. 10, p. 1745, Oct. 2022, doi: 10.3390/AGRICULTURE12101745.

[2] A. Rehman, T. Saba, M. Kashif, S. M. Fati, S. A. Bahaj, and H. Chaudhry, “A Revisit of Internet of Things Technologies for Monitoring and Control Strategies in Smart Agriculture,” *Agronomy 2022, Vol. 12, Page 127*, vol. 12, no. 1, p. 127, Jan. 2022, doi: 10.3390/AGRONOMY12010127.

[3] N. Rane, S. Choudhary, and J. Rane, “Artificial Intelligence (AI) and Internet of Things (IoT) – Based Sensors for Monitoring and Controlling in Architecture, Engineering, and Construction: Applications, Challenges, and Opportunities,” *SSRN Electronic Journal*, Nov. 2023, doi: 10.2139/SSRN.4642197.

[4] M. N. Mowla, N. Mowla, A. F. M. S. Shah, K. M. Rabie, and T. Shongwe, “Internet of Things and Wireless Sensor Networks for Smart Agriculture Applications: A Survey,” *IEEE Access*, vol. 11, pp. 145813–145852, 2023, doi: 10.1109/ACCESS.2023.3346299.

[5] K. Bhatt, C. Agrawal, and A. M. Bisen, “A Review on Emerging Applications of IoT and Sensor Technology for Industry 4.0,” *Wirel Pers Commun*, vol. 134, no. 4, pp. 2371–2389, Feb. 2024, doi: 10.1007/S11277-024-11054-X/METRICS.

[6] Z. Wu, K. Qiu, and J. Zhang, “A Smart Microcontroller Architecture for the Internet of Things,” *Sensors 2020, Vol. 20, Page 1821*, vol. 20, no. 7, p. 1821, Mar. 2020, doi: 10.3390/S20071821.

[7] M. Talal *et al.*, “Smart Home-based IoT for Real-time and Secure Remote Health Monitoring of Triage and Priority System using Body Sensors: Multi-driven Systematic Review,” *J Med Syst*, vol. 43, no. 3, pp. 1–34, Mar. 2019, doi: 10.1007/S10916-019-1158-Z/METRICS.

[8] R. Cheour, S. Khriji, M. Abid, and O. Kanoun, “Microcontrollers for IoT: Optimizations, Computing Paradigms, and Future Directions,” *IEEE World Forum on Internet of Things, WF-IoT 2020 - Symposium Proceedings*, Jun. 2020, doi: 10.1109/WF-IOT48130.2020.9221219.

[9] L. O. Aghenta and M. T. Iqbal, “Design and implementation of a low-cost, open source IoT-based SCADA system using ESP32 with OLED, ThingsBoard and MQTT protocol,” *AIMS Electronics and Electrical Engineering*, vol. 4, no. 1, pp. 57–86, Dec. 2019, doi: 10.3934/ELECTRENG.2020.1.57.

[10] M. A. Navarrete-Sanchez, R. Olivera-Reyna, R. Olivera-Reyna, R. J. Perez-Chimal, and J. U. Munoz-Minjares, “IoT-Based Classroom Temperature Monitoring and Missing Data Prediction Using Raspberry Pi and ESP32,” *Journal of Robotics and Control (JRC)*, vol. 6, no. 1, pp. 234–245, Jan. 2025, doi: 10.18196/JRC.V6I1.24345.

p-ISSN : 2087-0868 e-ISSN : 2598-9707

[11] D. Witczak and S. Szymoniak, "Review of Monitoring and Control Systems Based on Internet of Things," *Applied Sciences 2024, Vol. 14, Page 8943*, vol. 14, no. 19, p. 8943, Oct. 2024, doi: 10.3390/APP14198943.

[12] P. D. P. Adi, A. Kristiyanto, A. Sopandi, I. Nirmala, M. Tahir, and E. Y. Wijaya, "Performance Evaluation of Raspberry Pi Pico for Internet of Things and Its Analysis," *Proceedings - ICE3IS 2024: 4th International Conference on Electronic and Electrical Engineering and Intelligent System: Leading-Edge Technologies for Sustainable Societies*, pp. 7–12, 2024, doi: 10.1109/ICE3IS62977.2024.10775579.

[13] L. O. Aghenta and M. T. Iqbal, "Low-Cost, Open Source IoT-Based SCADA System Design Using Thinger.IO and ESP32 Thing," *Electronics 2019, Vol. 8, Page 822*, vol. 8, no. 8, p. 822, Jul. 2019, doi: 10.3390/ELECTRONICS8080822.

[14] C. Hendry and D. Manongga, "IMPLEMENTATION OF MULTI-NODE SENSOR DATA DELIVERY USING THE MASTER-SLAVE METHOD IN LORA COMMUNICATION," *Journal of Technology Informatics and Engineering*, vol. 3, no. 2, pp. 117–137, Aug. 2024, doi: 10.51903/JTIE.V3I2.179.

[15] R. Krishnamurthi, A. Kumar, D. Gopinathan, A. Nayyar, and B. Qureshi, "An Overview of IoT Sensor Data Processing, Fusion, and Analysis Techniques," *Sensors 2020, Vol. 20, Page 6076*, vol. 20, no. 21, p. 6076, Oct. 2020, doi: 10.3390/S20216076.

[16] O. Uviase and G. Kotonya, "IoT Architectural Framework: Connection and Integration Framework for IoT Systems," *Electronic Proceedings in Theoretical Computer Science, EPTCS*, vol. 264, pp. 1–17, Feb. 2018, doi: 10.4204/EPTCS.264.1.

[17] D. Serpanos and M. Wolf, "Internet-of-things (IoT) systems: Architectures, algorithms, methodologies," *Internet-of-Things (IoT) Systems: Architectures, Algorithms, Methodologies*, pp. 1–95, Nov. 2017, doi: 10.1007/978-3-319-69715-4/COVER.

[18] A. Joice *et al.*, "Applications of Raspberry Pi for Precision Agriculture—A Systematic Review," *Agriculture 2025, Vol. 15, Page 227*, vol. 15, no. 3, p. 227, Jan. 2025, doi: 10.3390/AGRICULTURE15030227.

[19] N. Khairiah *et al.*, "Solar Power IoT Based Smart Agriculture System Using NodeMCU ESP32," *Progress in Engineering Application and Technology*, vol. 5, no. 1, pp. 95–102, Jun. 2024, doi: 10.30880/peat.2024.05.01.010.

[20] Q. Alfalouji *et al.*, "IoT Middleware Platforms for Smart Energy Systems: An Empirical Expert Survey," *Buildings 2022, Vol. 12, Page 526*, vol. 12, no. 5, p. 526, Apr. 2022, doi: 10.3390/BUILDINGS12050526.