Studi Kinerja Transmisi Data Menggunakan ESP32 dan Raspberry Pi Pico Berbasis Simulasi Wokwi

[1]Arif Fahmi, [2]Indra Kurniawan
[12]Teknologi Rekayasa Komputer, Politeknik Negeri Banyuwangi, Banyuwangi
[1]ariffahmi@poliwangi.ac.id, [2] indrakurniawan@poliwangi.ac.id

Abstract - The development of the Internet of Things (IoT) has increased rapidly, and microcontrollers play a central role in the control and data communication in the IoT ecosystem. This research compares the data transmission performance between two popular microcontrollers, ESP32 and Raspberry Pi Pico, using the Wokwi simulation platform. The ESP32 is known for its Wi-Fi and Bluetooth connectivity, while the Raspberry Pi Pico with its RP2040 processor offers solid performance at an affordable price. In this context, MQTT is used as the communication protocol for data delivery with low latency and minimal bandwidth. This research includes three stages: system design, schematic design, and system testing. The scheme used includes DHT22 and LDR sensors, and the process of monitoring data to the Thingspeak server. The simulation results show that both microcontrollers can manage the data transmission well, but the Raspberry Pi Pico shows a faster transmission delay time than the ESP32. The average delay time for light intensity, temperature, and humidity parameters on the Raspberry Pi Pico was 2 minutes 36 seconds, 3 minutes 7 seconds, and 3 minutes 49 seconds, respectively, while the ESP32 had a longer delay time. This study provides insight into the performance of both microcontrollers in the context of the IoT and can help developers choose a suitable microcontroller for their applications.

Keywords — DHT22 Sensors, Data Trasnmission Performance, Esp32, Internet of things, LDR Sensors, MQTT Protocol, Raspberry Pi Pico, Wokwi Simulation Platform

Abstrak*—Perkembangan Internet of Things (IoT) telah mengalami peningkatan pesat, dengan mikrokontroler memiliki peran sentral dalam pengendalian dan komunikasi data dalam ekosistem IoT. Penelitian ini membandingkan kinerja transmisi data antara dua mikrokontroler populer, ESP32 dan Raspberry Pi Pico, menggunakan platform simulasi Wokwi. ESP32 dikenal dengan konektivitas Wi-Fi dan Bluetooth-nya, sementara Raspberry Pi Pico, dengan prosesor RP2040, menawarkan performa solid dengan harga terjangkau. Dalam konteks ini, MQTT digunakan sebagai protokol komunikasi untuk pengiriman data dengan latensi rendah dan bandwidth minimal. Penelitian ini melibatkan tiga tahap: perancangan sistem, perancangan skematik rangkaian, dan pengujian sistem. Skema yang digunakan melibatkan sensor DHT22 dan LDR, serta proses monitoring data ke server Thingspeak. Hasil simulasi menunjukkan bahwa kedua mikrokontroler dapat mengelola transmisi data dengan baik, tetapi Raspberry Pi Pico menunjukkan waktu delay transmisi yang lebih cepat dibandingkan ESP32. Rata-rata waktu delay untuk parameter intensitas cahaya, suhu, dan kelembapan udara pada Raspberry Pi Pico adalah masing-masing 2 menit 36 detik, 3 menit 7 detik, dan 3 menit 49 detik, sedangkan ESP32 memiliki waktu delay yang lebih lama. Studi ini memberikan wawasan tentang efisiensi kedua mikrokontroler dalam konteks IoT dan dapat membantu pengembang dalam memilih mikrokontroler yang sesuai untuk aplikasi mereka.

Kata Kunci— Esp32, Internet of Things, Protokol MQTT, Performa transmisi data, Raspberry Pi Pico, Sensor DHT dan LDR, Simulator Wokwi

I. Pendahuluan

Perkembangan konsep Internet of Things (IoT) terus meningkat seiring perkembangan waktu. IoT mengacu pada jaringan perangkat yang saling terhubung dan dapat berkomunikasi satu sama lain melalui internet [1][2]. Untuk mendukung ekosistem IoT, pemilihan perangkat keras yang tepat sangat krusial, terutama mikrokontroler yang menjadi pusat pengendalian dan komunikasi data. Mikrokontroler sebagai inti dari banyak aplikasi IoT, memainkan peran penting dalam pengelolaan data dan kontrol perangkat. Dalam konteks ini, ESP32 dan Raspberry Pi Pico muncul sebagai pilihan populer yang menawarkan kombinasi kemampuan, fleksibilitas, dan harga yang terjangkau [3][4]. ESP32 dikenal dengan kemampuannya untuk terhubung ke Wi-Fi dan Bluetooth, sementara Raspberry Pi Pico dengan prosesor RP2040-nya menawarkan performa yang solid dengan harga yang sangat kompetitif. Kedua mikrokontroler ini dapat digunakan secara efektif dalam sistem IoT, masing-masing dengan keunggulan dan karakteristik uniknya. Pada penelitian sebelumnya telah dilakukan analisi transmisi data internet of things menggunakan esp32 dengan parameter delay yang diujikan [5]. Adapun juga pada penelitian lain telah dilakukan pengujian aplikasi internet of things dengan rasberry pi pico untuk monitoring system kesehetan namun dalam data jurnal tersebut tidak dilakukan pengujian transmisi data[6]. Dalam implementasi sistem IoT, komunikasi antar perangkat menjadi aspek yang sangat penting. MQTT (*Message Queuing Telemetry Transport*) adalah protokol komunikasi yang sangat populer dalam lingkungan IoT karena efisiensinya dalam mengirimkan pesan dengan latensi rendah dan bandwidth rendah [7][8]. MQTT dirancang untuk berfungsi dalam jaringan yang tidak stabil dan sering digunakan dalam aplikasi yang memerlukan komunikasi yang cepat dan andal antar perangkat. Kebutuhan sarana dan prasarana yang

diperlukan untuk peracangan system Internet of Things ini tentunya membutuhkan biaya yang tidak sedikit, bagi beberapa kalangan dalam memenuhi kebutuhan seperti perangkat keras ataupun perangkat lunak untuk pembelajaran internet of things ini masih menjadi masalah. Dalam penelitian ini diperlukan sofware emulator untuk mengatasi masalah keterbatasan perangkat keras maupun perangkat lunak sehingga studi perbandingan kinerja transmisi data Internet of Things dapat terus berlangsung secara virtual. Wokwi adalah salah satu platform simulasi yang memungkinkan pengguna untuk membangun dan menguji skema elektronik secara online [9][10]. Dengan menggunakan Wokwi, kita dapat menganalisis kinerja transmisi data IoT menggunakan mikrokontroler ESP32 dan Raspberry Pi Pico dalam konteks simulasi.

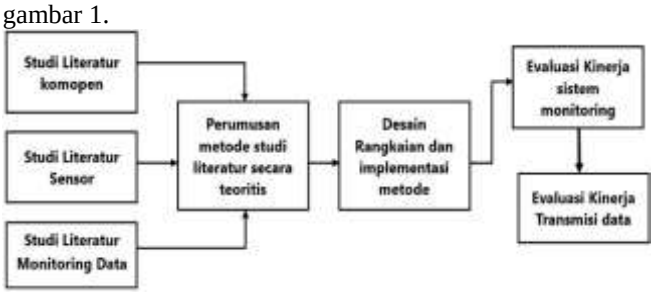
Studi ini bertujuan untuk melakukan perbandingan kinerja transmisi data antara mikrokontroler ESP32 dan Raspberry Pi Pico menggunakan platform simulasi Wokwi. Fokus utama dari penelitian ini adalah untuk mengevaluasi efisiensi dan efektivitas kedua mikrokontroler dalam mengelola dan mentransmisikan data dalam skenario IoT. Analisis ini akan mempertimbangkan berbagai parameter kinerja, termasuk kecepatan transmisi data, dan stabilitas komunikasi. Dengan melakukan studi perbandingan ini, diharapkan dapat memberikan panduan yang berguna bagi pengembang dan peneliti dalam memilih mikrokontroler yang sesuai untuk aplikasi IoT mereka.

II. Metode Penelitian

Pada metode penelitian ini penulis membagi menjadi beberapa tahap. Tahapan pertama yaitu perancangan sistem, kedua tahap perancangan skematik rangkaian dan ketiga tahap pengujian sistem

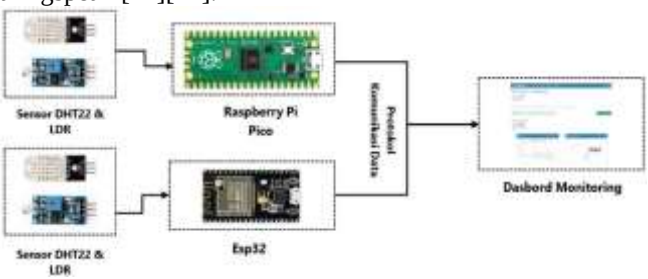
A. Perancangan Sistem

Pada perancangan sistem ini penulis melakukan pemetaan metodologi studi perbandingan kinerja transmisi data pada mikrokontroler Esp32 dengan Raspberry Pi Pico sebagaimana gambar 1.



Gambar 1. Metodologi penelitian studi perbandingan kinerja transmisi data mikrokontroler Esp32 dan Raspberry Pi Pico

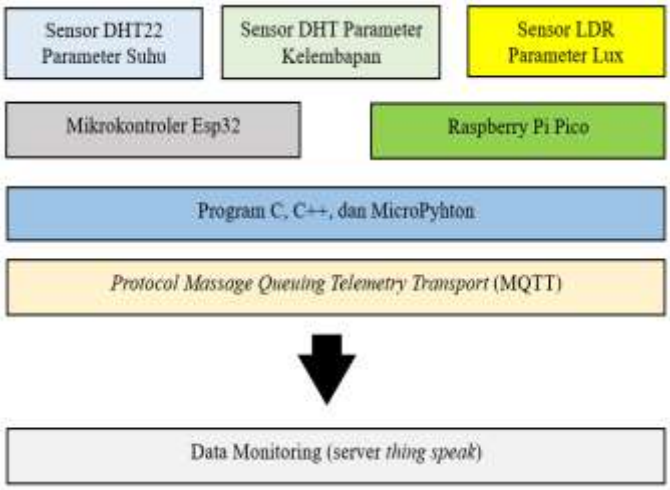
Tahapan pertama pada penelitian yang dilakukan dengan mempelajari *datasheet* dari masing-masing mikrokontroler Esp32 dan raspberry pi pico. Pada bagian studi literature sensor penulis menggunakan sensor DHT-22 dan sensor LDR (light dependent resistor), meskipun pada beberapa penelitian sebelumnya yang terkait terdapat penggunaan sensor yang lebih komplek, namun pada penelitian ini dibatasi dengan penggunaan sensor yang sederhana yaitu sensor DHT22 dan LDR karena penulis berfokus pada pengujian perbandingan kinerja transmisi data untuk mikrokontroler Esp32 dan raspberry pi pico. Pada penelitian ini juga melakukan studi perancangan perangkat lunak yang digunakan untuk proses monitoring data yang dikirimkan dari perangkat ke server thingspeak [11][12].



Gambar 2. Pemanfaatan sensor DHT22 dan LDR pada proses sensing

Selanjutnya tahap perancangan skematik rangkaian berbasis dengan simulator wokwi, selanjutnya dilakukan tahap pengujian kinerja sistem monitoring ke server thingspeak untuk berikutnya dilakukan pengujian dan perbandingan kinerja transmisi data yang dalam hal ini parameter delay yang akan dibandingkan.

Pada penelitian ini penulis juga membuat pemetaan arsitektur sistem yang dirancang sebagaimana Gambar 3 [13].



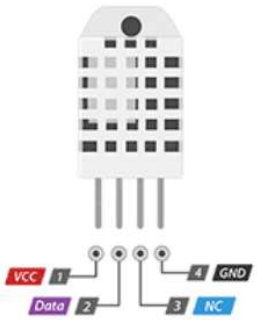
Gambar 3. Arsitektur Sistem

Sistem pada penelitian ini menggunakan sensor DHT22 dan LDR yang digunakan untuk melakukan *sensing. Controller* yang digunakan adalah mikrokontroler Esp32 dan raspberry pi pico. Bahasa pemrograman yang digunakan adalah kombinasi Bahasa C, C++ dan Micropyhton. Protocol komunikasi data dilakukan dari *controller* ke server *Thingspeak* adalah protocol MQTT.

B. Perancangan Skematik Rangkaian

Sebelum melakukan perancangan skematik rangkaian penulis melakukan pemetaan spesifikasi masing masing komponen sebagaimana berikut.

Sensor DHT22



Gambar 4. Skematik Sensor DHT22 [14]

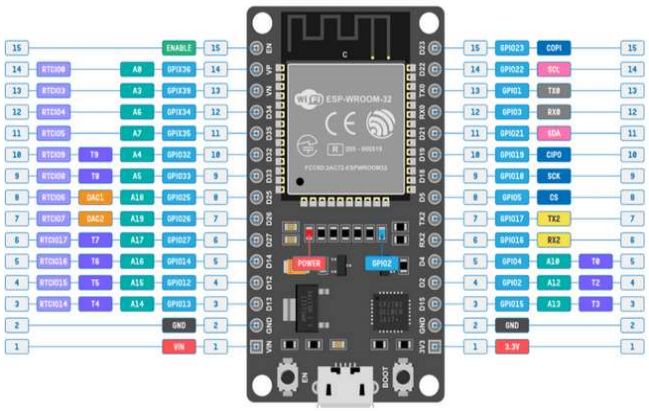
Sensor DHT22 memiliki spesifikasi pin sebanyak 4 pin yaitu pin Vcc berfungsi sebagai pin power (sumber daya), pin data sebagai pin output data parameter sensor, pin NC sebagai pin tanpa koneksi yang terhubung dengan ground, dan pin GND sebagai grounding.

Sensor LDR

Gambar 5. Skematik Sensor LDR [15]

Sensor LDR memliki spesifikasi pin sebanyak 4 pin yaitu pin Vcc sebagai pin power (sumber daya), pin A0 berfungsi sebagai output data sensor pada pin analog, pin D0 berfungsi sebagai output data sensor pada pin digital dan pin Gnd berfungsi sebagai grounding.

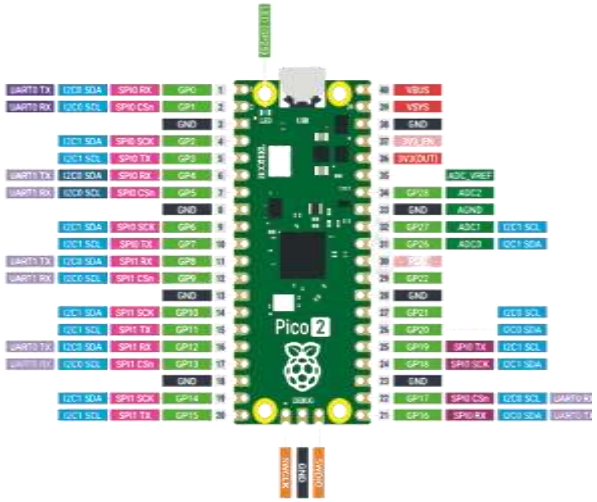
Mikrokontroler ESP32



Gambar 6. Skematik Mikrokontroler Esp32 [16]

Mikrokontroler Esp32 secara garis besar memiliki spesifikasi *peripheral* diantaranya, 18 kanal ADC (Analog-to-Digital Converter), 3 antarmuka SPI, 3 antarmuka UART, 2 antarmuka I2C, 16 kanal output PWM, 2 kanal DAC (Digital to Analog Converter), 2 antarmuka I2S, serta 10 pin GPIO sensor kapasitif.

Raspberry Pi Pico

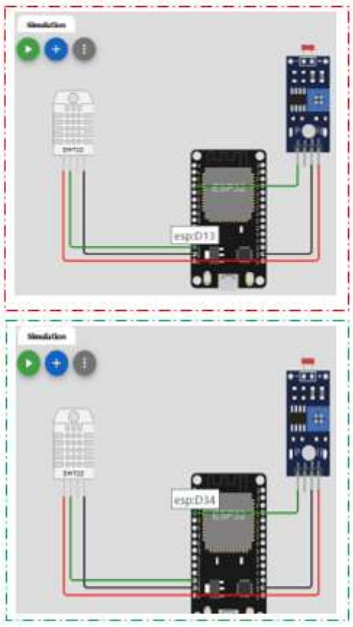


Gambar 7. Skematik Mikrokontroler raspberry Pi Pico [17]

Mikrokontroler Raspberry pi Pico secara garis besar memiliki spesifikasi *peripheral* diantaranya, 3 kanal ADC 12-bit, 2 antarmuka SPI, 2 antarmuka UART, 1 kanal PWM (pulse width modulation), 2 antarmuka I2C, serta 26 pin GPIO sensor kapasitif.

Pada perancangan skematik rangkaian penulis merancang sistem internet of things menggunakan simulator wokwi sebagaimana beikut.

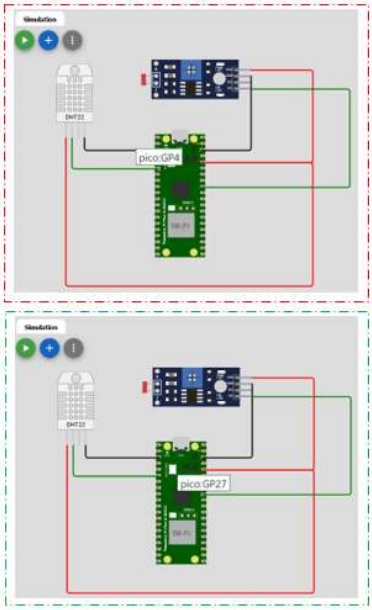
Desain Rangkaian IoT dengan Mikrokontroler ESP 32



Gambar 8. Desain Rangkaian IoT dengan Mikrokontroler ESP32

Pada gambar 8 menunjukan desain rangkaian IoT mikrokontroler ESP32 yang terintegrasi dengan sensor DHT22 dengan mensetting pin data pada sensor ke pin GPIO13 (D13) Esp32 sedangkan pin A0 sensor LDR diintegrasikan dengan pin GPIX34 (D34).

Desain Rangkaian IoT dengan Raspberry Pi Pico

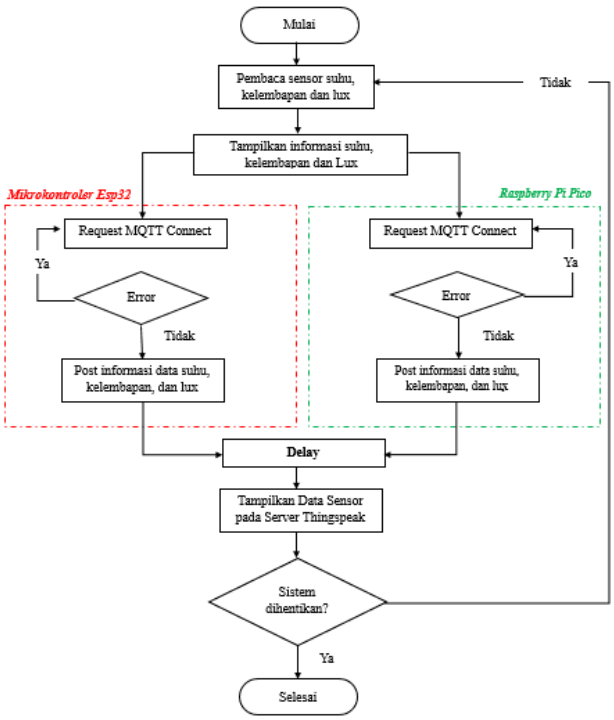


Gambar 9. Deain rangkaian IoT dengan Raspberry Pi Pico

Pada gambar 9 menunjukan desain rangkaian IoT mikrokontroler Raspberry pi pico yang terintegrasi dengan sensor DHT22 dengan mensetting pin data pada sensor ke pin GP4 Raspberry pi pico sedangkan pin A0 sensor LDR diintegrasikan dengan pin GP27.

C. Pengujian Sistem

Pada pengujian sistem penulis melakukan dengan pemetaan flowchat sebagaimana gambar 10.



Gambar 10. Pemetaan pengujian sistem IoT

Pada pemetaan pengujian sistem tersebut penulis membuat pemetaan diagram alir untuk mengetahui bagiamana perbandingan kinerja transmisi data internet of things antara mikrokontroler Esp32 dengan raspberry pi pico yang dalam hal ini parameter yang diuji adalah delay dalam pengiriman data ke server thingspeak.

III. Hasil dan Pembahasan

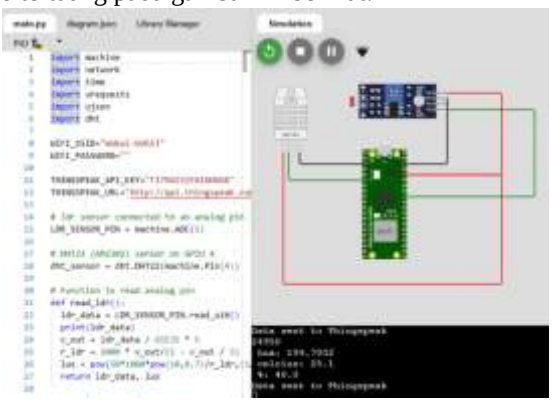
A. Hasil Simulasi Rangkaian

Berikut tampilan hasil simulasi rangkaian IoT mikrokontroler Esp32 pada simulator wokwi yang tertuang pada gambar 11.

Gambar 11. Hasil Simulasi Rangkaian Mikrokontroler Esp32

Berdasarkan gambar 11 menunjukkan hasil simulasi rangkaian mikrokontroler Esp32 berjalan dengan baik dibuktikan dengan tampilan serial monitor berwarna hitam yang menampilkan hasil parameter sensor suhu sebesar 21.2C^0 parameter kelembapan udara sebesar 69.5% dan intensitas cahaya 559.8505 lux.

Adapun hasil simulasi rangkaian IoT mikrokontroler Raspberry pi pico tertuang pada gambar 12 berikut.

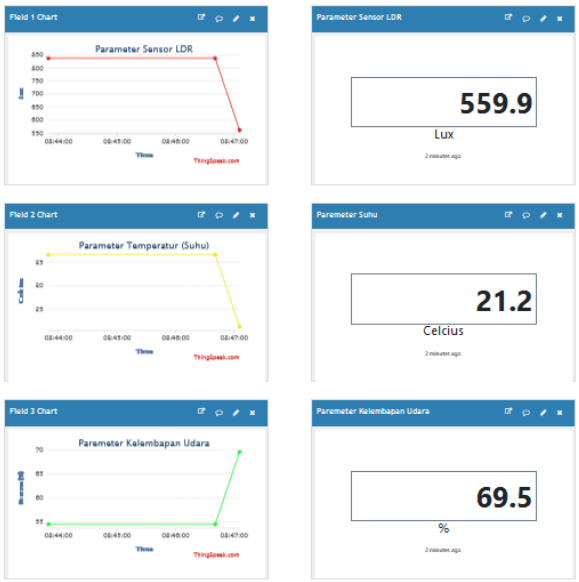


Gambar 12. Hasil Simulasi Rangkaian Raspberry pi pico

Berdasarkan gambar 12 menunjukkan hasil simulasi rangkaian mikrokontroler raspberry pi pico berjalan dengan baik dibuktikan dengan tampilan serial monitor berwarna hitam yang menampilkan hasil parameter sensor suhu sebesar 25.1C^0 parameter kelembapan udara sebesar 40% dan intensitas cahaya 199.7002 lux.

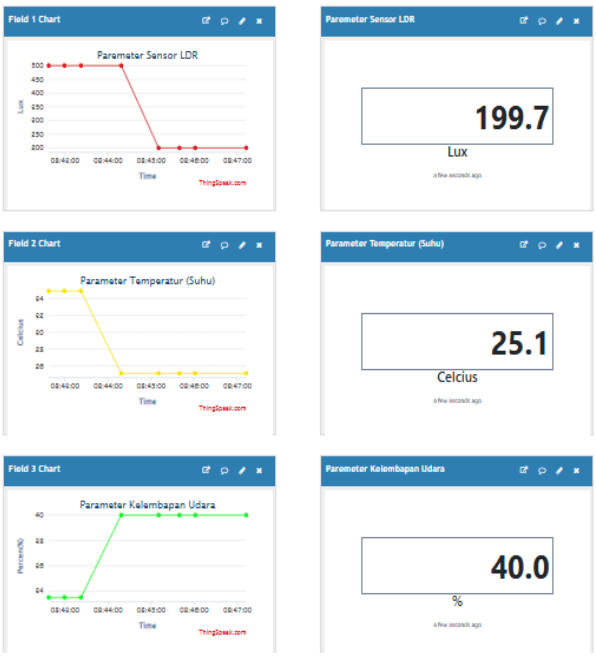
B. Hasil Post Data Ke Server ThingSpeak

Pada sub bab ini penulis memaparkan hasil post data parameter sensor menuju server thingspeak sebagai validasi data dari simulasi wokwi rangkaian sebagaimana tertuang pada gambar 13 dan 14.



Gambar 13. Grafik parameter sensor dengan mikrokontroler Esp32 pada server Thingspeak.

informasi pada gambar 13 diperoleh data yang sama dengan hasil simulasi rangkaian mikrokontroler Esp32 pada gambar 11 yaitu hasil parameter sensor suhu sebesar 21.2C^0 parameter kelembapan udara sebesar 69.5% dan intensitas cahaya 559.8505 lux.



Gambar 14. Grafik parameter sensor dengan mikrokontroler Esp32 pada server Thingspeak

pada gambar 14 diperoleh data yang sama dengan hasil simulasi rangkaian mikrokontroler Esp32 pada gambar 12 yaitu hasil parameter sensor suhu sebesar 25.1C^0 parameter kelembapan udara sebesar 40.0% dan intensitas cahaya 199.7002 lux.

pada gambar 11,12,13,14 menunjukkan parameter data sensor dapat terkirim ke server dengan baik dalam hal ini dapat dinyatakan pada tabel 1.

Tabel 1. Validasi transmisi data sensor simulasi rangkaian ke server

No	*Parameter sensor*	*Hasil Simulasi Rangkain*		*Data tampilan Thingspeak*		*Persentase*	*Validasi*
		Esp 32	*Raspberry pi pico*	*Esp 32*	*Raspberry pi pico*		
1	Suhu (C^0)	21.2	25.1	21.2	25.1	100%	✓
2	Kelembapan (%)	69.5	40	69.5	40	100%	✓
3	Intensitas cahaya (lux)	559.8505	199.7002	559,9	199.7	99,99%	✓

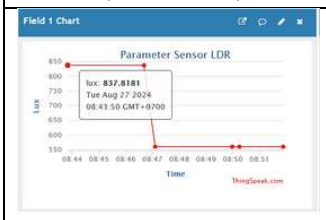
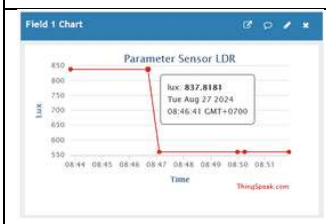
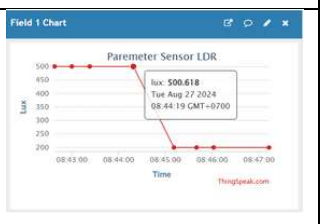
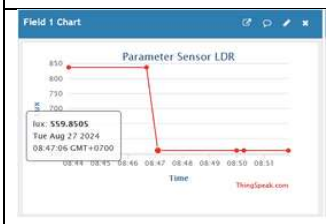
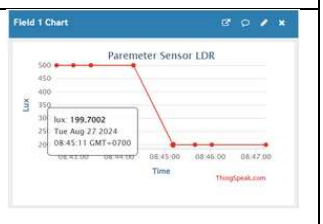
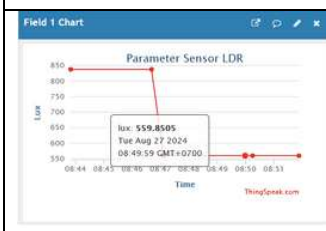
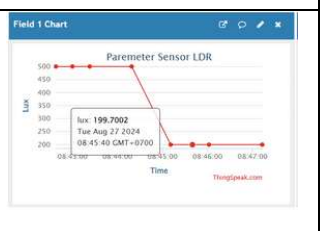
C. Hasil Perbandingan Transmisi Data

Pada sub bab ini penulis memaparkan hasil perbandingan transmisi data dengan parameter delay pada mikrokontroler esp 32 dan mikrokontroler raspberry pi pico.

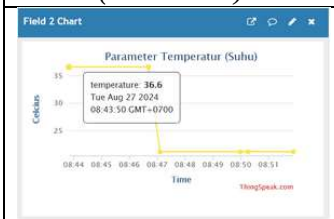
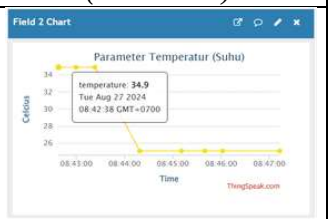
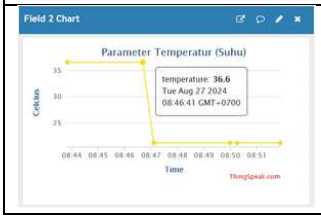
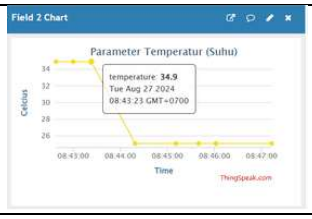
Gambar 15. Perbandingan data grafik sensor pada thingspeak

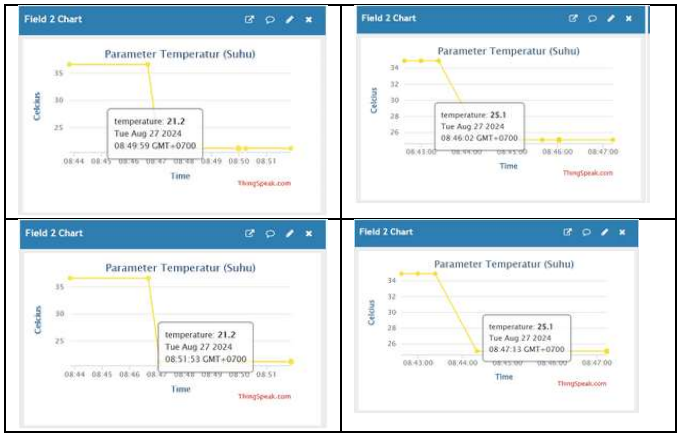
pada gambar 15 menujukkan perbandingan grafik masing-masing parameter sensor dimana grafik sebelah kiri menunjukkan hasil parameter sensor pada mikrokontroler Esp32 dan grafik sebelah kanan menunjukkan hasil parameter sensor pada raspberry pi pico, pada gambar tersebut belum dapat menganalisa perbedaan yang lebih mendalam terkait nilai parameter delay dari masing-masing mikrokontroler sehingga dipaparkan pada keterangan berikut.

Tabel 2. Perbandingan waktu transmisi data mikrokontroler Esp32 dengan raspberry pi pico pada sensor ldr (parameter lux)

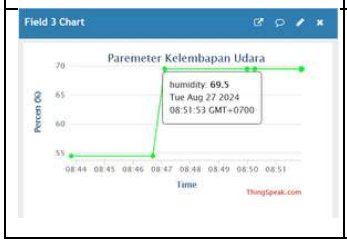
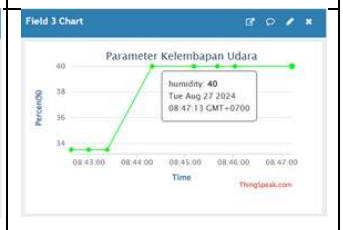
Mikrokontroler Esp32 (sensor LDR)	*Raspberry Pi Pico (sensor LDR)*
	
	
	
	

Tabel 3. Perbandingan waktu transmisi data mikrokontroler Esp32 dengan raspberry pi pico pada sensor DHT22 (parameter suhu)

Mikrokontroler Esp32 (sensor LDR)	*Raspberry Pi Pico (sensor LDR)*
	
	



Tabel 4. Perbandingan waktu transmisi data mikrokontroler Esp32 dengan raspberry pi pico pada sensor DHT22 (parameter kelembapan udara)

Mikrokontroler Esp32 (sensor LDR)	*Raspberry Pi Pico (sensor LDR)*
	
	
	
	

pada data grafik pada table 2, 3 dan 4 menujukkan masing masing parameter nilai dan waktu tempuh (delay) saat data terbaca pada server thingspeak, berikut pemetaan perbandingan transmisi data (delay) pada mikrokonrtoler Esp32 dengan raspberry pi pico.

Tabel 5. Perbandingan waktu transmisi data mikrokontroler Esp32 dengan raspberry pi pico pada masing masing parameter sensor

Parameter sensor	Mikrokontroler Esp32		Raspberry Pi Pico		Perbandingan delay	Lebih cepat
	Data	waktu	Data	Waktu		
Intensitas Cahaya (Lux)	837	08:43:50	500	08:42:38	1 menit 12 detik	Pi pico
	837	08:46:41	500	08:44:19	2 menit 22 detik	Pi pico
	559	08:47:06	199	08:45:11	1 menit 55 detik	Pi pico
	559	08:49:59	199	08:45:40	4 menit 19 detik	Pi pico
Suhu (C^0)	36.6	08:43:50	34.9	08:42:38	1 menit 12 detik	Pi pico
	36.6	08:46:41	34.9	08:43:23	2 menit 18 detik	Pi pico
Kelembapan Udara (%)	21.2	08:49:59	25.1	08:46:02	3 menit 57 detik	Pi pico
	21.2	08:51:53	25.1	08:47:13	4 menit 40 detik	Pi pico
Kelembapan Udara (%)	54.5	08:43:50	33.5	08:42:38	1 menit 12 detik	Pi pico
	54.5	08:46:41	33.5	08:43:23	3 menit 18 detik	Pi pico
	69.5	08:47:06	40	08:44:19	2 menit 47 detik	Pi pico
	69.5	08:51:53	40	08:47:13	4 menit 40 detik	Pi pico

Pada tabel 5 diperoleh informasi perbandingan parameter delay saat mengirim data ke server thingspeak antara mikrokontroler Esp32 dengan raspberry pi pico dengan hasil delay yang lebih cepat pada mikrokontroler raspberry pi pico. Dengan data rata-rata (±) waktu delay pada parameter intensitas cahaya (lux) sebesar 2 menit 36 detik, pada parameter suhu (C^0) sebesar 3 menit 7 detik, dan pada parameter kelembapan udara (%) sebesar 3 menit 49 detik.

IV. Kesimpulan

Kesimpulan pada penelitian dijabrakan dalam beberapa point diantanya

1. Data yang dihasilkan dari simulasi rangkaian menggunakan mikrokontroler ESP32 dan Raspberry Pi Pico berhasil dikirimkan dan divalidasi dengan baik pada server ThingSpeak, data yang ditampilkan pada menunjukkan integritas data yang tinggi dengan persentase validasi mencapai hampir 100%.
2. Hasil perbandingan waktu transmisi data antara mikrokontroler ESP32 dan Raspberry Pi Pico menunjukkan bahwa Raspberry Pi Pico memiliki waktu transmisi yang lebih cepat dibandingkan ESP32. Untuk parameter intensitas cahaya (lux), Raspberry Pi Pico lebih cepat dengan rata-rata waktu delay sebesar 2 menit 36 detik. Untuk parameter suhu (C0) dan kelembapan udara (%), Raspberry Pi Pico juga lebih unggul dengan rata-rata waktu delay masing-masing sebesar 3 menit 7 detik dan 3 menit 49 detik.

V. Daftar Pustaka

[1] S. Megawati, "Pengembangan Sistem Teknologi Internet of Things Yang Perlu Dikembangkan Negara Indonesia," *J. Inf. Eng. Educ. Technol.*, vol. 5, no. 1, pp. 19–26, 2021, doi: 10.26740/jieet.v5n1.p19-26.

[2] I. U. Turyadi, "Analisa Dukungan Internet of Things (IoT) terhadap Peran Intelejen dalam Pengamanan Daerah Maritim Indonesia Wilayah Timur," *J. Teknol. dan Manaj. Inform.*, vol. 7, no. 1, pp. 29–39, 2021, doi: 10.26905/jtmi.v7i1.6040.

[3] S. O. Ooko, "A Comparison of Arduino, Raspberry Pi and ESP8266 Boards Security of wireless networks View project Internet of Things View project," no. December, pp. 3–5, 2019, [Online]. Available: https://www.researchgate.net/publication/337707190

[4] A. K. Permana and A. Rachmawan, "Studi Komparasi Platform Open-Source Internet of Things," *J. Teknol. dan Manaj.*, vol. 21, no. 1, pp. 43–48, 2023, doi: 10.52330/jtm.v21i1.38.

[5] M. I. Fatchu Reza, "Analisis Kinerja Komunikasi Data Pada Model Taman Cerdas," *J. Pendidik. Sains dan Komput.*, vol. 2, no. 2, pp. 297–300, 2022, [Online]. Available: https://doi.org/10.47709/jpsk.v2i2.1755

[6] K. Kumar, S. Bhaumik, and S. L. Tripathi, "Health monitoring system," *Electron. Devices, Circuits, Syst. Biomed. Appl. Challenges Intell. Approach*, vol. 4, no. 2, pp. 461–480, 2021, doi: 10.1016/B978-0-323-85172-5.00018-6.

[7] R. F. Pratama, R. S. R. Wicaksono, and A. N. Pramudhita, "Perancangan Dan Implementasi Protokol Mqtt Pada Sistem Parkir Cerdas Berbasis Iot," *J. Inform. dan Tek. Elektro Terap.*, vol. 11, no. 3, pp. 475–483, 2023, doi: 10.23960/jitet.v11i3.3191.

[8] A. Kurnianto, J. Dedy Irawan, and F. Xaverius Ariwibisono, "Penerapan Iot (Internet of Things) Untuk Controlling Lampu Menggunakan Protokol Mqtt Berbasis Web," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 6, no. 2, pp. 1153–1161, 2023, doi: 10.36040/jati.v6i2.5393.

[9] L. Khakim, E. Budihartono, A. Rakhman, and A. Sutanto, "Pemanfaatan Aplikasi Wokwi sebagai Media Pembelajaran Mikrokontroler Berbasis Simulator di SMK Dinamika Kota Tegal," vol. 7, no. 2, pp. 385–391, 2024.

[10] E. Sabara and Wahyudi, "Desain Dan Implementasi Media Pembelajaran Mikrokontroler Berbasis Hybrid Learningmenggunakan Wokwi Simulation," *J. MEDIA Elektr.*, vol. 19, no. 3, pp. 186–193, 2022.

[11] M. Fauzan and N. Nurdin, "Implementasi Thingspeak Sebagai Database Pada Alat Deteksi Banjir Menggunakan Esp32," *CONTEN Comput. Netw. Technol.*, vol. 4, no. 1, pp. 66–70, 2024, doi: 10.31294/conten.v4i1.3596.

[12] A. Agung and G. Ekayana, “Implementasi Sipratu Menggunakan Platform,” vol. 8, pp. 237–248, 2019.
[13] A. Wardhana *et al.*, “Arsitektur dan standarisasi internet of things (iot),” *E-Commerce Dan Internet Things*, no. May, pp. 197–210, 2023.
[14] UTMEL ELECTRONIC, “DHT11 vs DHT22: Datasheet, Pinout, and Features.” Accessed: Mar. 03, 2025. [Online]. Available: https://www.utmel.com/components/dht11-vs-dht22-datasheet-pinout-and-features?id=1148
[15] D. Ables, “DIYables LDR Light Sensor Module for Arduino, ESP32, ESP8266, Raspberry Pi, 4 pieces.” Accessed: Mar. 04, 2025. [Online]. Available: https://diyables.io/products/ldr-light-sensor-module
[16] CIRCUITSTATE, “Getting Started with Espressif ESP32 Wi-Fi & Bluetooth SoC using DOIT-ESP32-DevKit-V1 Development Board.” Accessed: Mar. 04, 2025. [Online]. Available: https://www.circuitstate.com/tutorials/getting-started-with-espressif-esp32-wifi-bluetooth-soc-using-doit-esp32-devkit-v1-development-board/
[17] R. Pi, “Pico-series Microcontrollers.” Accessed: Mar. 05, 2025. [Online]. Available: https://www.raspberrypi.com/documentation/microcontrollers/pico-series.html#pico-1-family