

ANALISIS KONSUMSI ENERGI NODE IOT BERBASIS FSM PADA PROTOKOL MQTT

Farrel Daffana, Uray Ristian, Hirzen Hasfani

Rekayasa Sistem Komputer, Universitas Tanjungpura

Jl. Jenderal Ahmad Yani, Bansir Laut, Kec. Pontianak Tenggara, Kota Pontianak

h1051211031@student.untan.ac.id

ABSTRAK

Perkembangan teknologi *Internet of Things* (IoT) mendorong kebutuhan sistem pemantauan yang handal dan hemat energi pada perangkat dengan sumber daya terbatas. Namun, minimnya pemahaman terhadap profil konsumsi energi perangkat IoT secara nyata menjadi tantangan dalam merancang sistem yang efisien. Penelitian ini bertujuan menganalisis konsumsi energi NodeMCU ESP32 yang terintegrasi dengan sensor *soil moisture* dan DHT11 menggunakan protokol komunikasi MQTT. Pendekatan *Finite State Machine* (FSM) diterapkan untuk mendefinisikan empat state operasional, yaitu *Disconnected*, *Connecting*, *Connected*, dan *Publishing*, sebagai kerangka pengukuran yang sistematis. Pengukuran dilakukan selama 12 siklus operasional dalam dua sesi, masing-masing berdurasi satu jam. Hasil menunjukkan bahwa state *Connected* mendominasi 93,75% dari total waktu operasional, sementara state *Publishing* mencakup 5,98%. Rata-rata konsumsi energi per siklus adalah 0,4304 Wh, dengan proyeksi konsumsi harian sebesar $\pm 10,33$ Wh pada operasi kontinu 24 jam. Sesi 2 memperlihatkan konsistensi lebih baik dengan range hanya 0,0150 Wh dibandingkan Sesi 1. Implementasi FSM terbukti menghasilkan profil konsumsi energi yang stabil dan dapat menjadi acuan teknis dalam perancangan sistem IoT hemat energi.

Kata kunci : konsumsi energi, MQTT, finite state machine, IoT

1. PENDAHULUAN

Internet of Things (IoT) merupakan teknologi yang memungkinkan perangkat sensor dan aktuator terhubung melalui jaringan untuk memantau dan mengendalikan proses secara *remote* [1].

Salah satu tantangan utama dalam pengembangan IoT adalah konsumsi energi yang belum optimal, terutama karena perangkat IoT umumnya bergantung pada sumber energi terbatas seperti baterai atau suplai listrik dengan efisiensi rendah. Dalam jaringan IoT berskala besar, manajemen daya yang buruk dapat meningkatkan biaya operasional, menurunkan efisiensi sistem, serta memperpendek umur perangkat keras [2].

ESP32 merupakan salah satu mikrokontroler yang banyak digunakan dalam implementasi IoT berkat fleksibilitas dan dukungan komunikasinya. Namun, perangkat ini memiliki variasi konsumsi daya yang dipengaruhi oleh kondisi operasional, seperti beban sensor dan aktivitas komunikasi data [3].

Konsumsi energi ESP32 sangat bergantung pada kondisi operasional yang sedang berjalan, di mana aktivitas Wi-Fi untuk keperluan transmisi data menjadi kontributor utama penggunaan daya pada perangkat ini. Oleh karena itu, pemahaman terhadap pola konsumsi energi ESP32 dalam setiap kondisi operasional menjadi hal yang krusial, khususnya dalam pengembangan sistem IoT yang menuntut efisiensi energi tinggi dan keandalan jangka panjang.

Protokol *Message Queuing Telemetry Transport* (MQTT) dirancang sebagai protokol *lightweight* yang cocok untuk perangkat dengan keterbatasan sumber daya, karena efisien dalam penggunaan *bandwidth* serta memiliki *overhead* komunikasi yang rendah. Hal

ini menjadikan MQTT lebih hemat energi dibanding protokol komunikasi lainnya [4].

Efisiensi daya dalam MQTT sangat dipengaruhi oleh konfigurasi jaringan serta strategi manajemen koneksi [5].

Dalam penelitian ini, pendekatan *Finite State Machine* (FSM) digunakan sebagai kerangka acuan dalam pembangunan prototipe, guna memastikan setiap kondisi operasional perangkat IoT terdefinisi secara eksplisit pada setiap tahapan komunikasi MQTT sehingga pengukuran konsumsi energi dapat dilakukan secara sistematis dan terukur. State operasional perangkat IoT dalam yang digunakan yaitu *Disconnected*, *Connecting*, *Connected*, dan *Publishing*. Dengan pendekatan ini, konsumsi energi dapat diukur dan dievaluasi pada setiap kondisi operasional. Misalnya, ketika *node* berkomunikasi langsung dengan *broker* MQTT, konsumsi energi saat *publishing* dapat dipetakan secara lebih terukur sebagai acuan evaluasi efisiensi sistem [6] [7].

Meskipun demikian, kajian yang secara khusus mengukur dan menganalisis konsumsi daya pada NodeMCU ESP32 yang terintegrasi dengan protokol MQTT dalam skenario operasional nyata masih terbatas. Sebagian besar penelitian yang ada belum memetakan secara detail bagaimana konsumsi energi terdistribusi di setiap state operasional MQTT pada platform ESP32, terutama ketika perangkat dilengkapi dengan sensor lingkungan seperti *soil moisture* dan DHT11 yang aktif secara bersamaan. Oleh karena itu, penelitian ini bertujuan menganalisis konsumsi energi NodeMCU ESP32 yang terintegrasi dengan sensor *soil moisture* dan DHT11 menggunakan komunikasi MQTT, dengan membangun prototipe berbasis FSM

dan mengukur konsumsi arus pada setiap state operasional menggunakan multimeter digital dalam skenario pengujian nyata. Hasil penelitian ini diharapkan dapat menjadi referensi dalam pengembangan sistem IoT yang lebih hemat energi, efisien, dan berkelanjutan.

2. TINJAUAN PUSTAKA

Pada tahap ini, peneliti melakukan kajian pustaka dengan mempelajari berbagai sumber informasi yang relevan dengan masalah yang dihadapi yaitu terkait konsumsi energi pada *node publisher* MQTT. Literatur yang dicari biasanya meliputi jurnal ilmiah, buku, artikel, laporan penelitian terdahulu, serta dokumen teknis yang terkait dengan topik utama. Tujuannya adalah untuk memahami landasan teori, metode yang digunakan di masa lalu, serta hasil yang diperoleh dari studi yang serupa. Studi literatur ini juga dapat membantu mengidentifikasi celah dalam penelitian sebelumnya yang menjadi dasar pengembangan sistem. Adapun tinjauan pustaka yang didapat seperti:

2.1. Penelitian Terdahulu

Beberapa penelitian sebelumnya telah mengkaji efisiensi energi pada sistem IoT. Akintade et al. mengembangkan model konsumsi daya untuk node pemantauan IoT berbiaya rendah berbasis ESP8266, dengan menganalisis perilaku konsumsi daya pada berbagai mode operasi perangkat guna mendukung perancangan sistem IoT yang lebih efisien [8]. Jeddou et al. melakukan karakterisasi empiris terhadap konsumsi energi dan latensi protokol MQTT yang berjalan di atas QUIC menggunakan perangkat komersial, dan menunjukkan bahwa MQTT berbasis QUIC menawarkan keunggulan tertentu dibandingkan implementasi berbasis TCP pada kondisi jaringan yang bervariasi [9]. Selain itu, penelitian Bordin et al. menekankan pentingnya analisis konsumsi energi dalam meningkatkan efisiensi dan keandalan sistem energi pintar berbasis IoT [10]. Ketiga penelitian tersebut secara kolektif menegaskan bahwa pemilihan strategi manajemen daya dan protokol komunikasi yang tepat berkontribusi signifikan terhadap peningkatan efisiensi dan keandalan sistem IoT secara keseluruhan.

2.2. Message Queuing Telemetry Transport

MQTT adalah protokol komunikasi berbasis publish/subscribe yang dirancang untuk transfer data yang ringan, efisien, dan andal dalam jaringan dengan bandwidth terbatas atau latensi tinggi. MQTT menggunakan mekanisme *publish-subscribe* yang memungkinkan komunikasi asinkron dan lebih efisien. Protokol ini menggunakan arsitektur *client-server*, di mana *broker* berfungsi sebagai perantara yang mengelola komunikasi antara *publisher* (pengirim pesan) dan *subscriber* (penerima pesan). MQTT bekerja dengan mekanisme topik, di mana *publisher* mengirimkan pesan ke topik tertentu, dan *subscriber*

yang berlangganan ke topik tersebut akan menerima pesan yang relevan.

Pendekatan ini memungkinkan komunikasi asinkron, sehingga perangkat tidak perlu terus-menerus melakukan polling untuk mendapatkan data baru, yang pada akhirnya dapat menghemat bandwidth dan konsumsi daya. Setiap topik dalam MQTT memiliki struktur hierarkis, memungkinkan pengorganisasian data yang lebih fleksibel dan memudahkan manajemen komunikasi dalam sistem IoT skala besar. MQTT dapat menangani ribuan klien jarak jauh menggunakan satu server[11].

2.3. Finite State Machine

FSM adalah algoritma komputasi yang digunakan untuk merepresentasikan sistem atau perangkat yang beroperasi dalam sejumlah keadaan (*state*) tertentu, dengan transisi antar-keadaan yang ditentukan oleh kondisi atau peristiwa spesifik [12]. FSM terdiri dari sejumlah komponen utama; kumpulan *state*, *input*, *output*, dan aturan transisi yang menghubungkan *state* satu dengan lainnya berdasarkan input yang diterima.

2.4. NodeMCU ESP32

NodeMCU ESP32 merupakan modul mikrokontroler berbasis chip ESP32 yang dikembangkan oleh Espressif Systems dan dirancang untuk aplikasi IoT dengan fungsionalitas yang sangat lengkap. NodeMCU sangat fungsional digunakan sebagai sumber pengedali dan pengolahan data. Salah satu keunggulan utama ESP32 adalah konektivitas nirkabelnya yang mendukung protokol WiFi 802.11 b/g/n untuk terhubung ke jaringan lokal atau internet, serta Bluetooth 4.2 dan BLE (*Bluetooth Low Energy*) untuk komunikasi jarak pendek. Mikrokontroler ESP-32 telah banyak digunakan dalam pembuatan sistem IoT yang murah [13]. Adapun Gambaran dari NodeMCU dapat dilihat pada Gambar 1.



Gambar 1. NodeMCU ESP32

2.5. DHT11

DHT11 adalah sensor suhu dan kelembapan yang banyak digunakan dalam berbagai proyek elektronik dan IoT karena harganya yang terjangkau dan kemudahan penggunaannya[14].

Sensor ini mampu membaca suhu dalam rentang 0°C hingga 50°C dengan akurasi $\pm 2^\circ\text{C}$, serta kelembapan relatif (RH) dalam rentang 20% hingga

90% dengan akurasi $\pm 5\%$. Adapun gambaran dari DHT11 yang digunakan dapat dilihat pada Gambar 2.



Gambar 2. Sensor DHT11

2.6. Sensor Soil moisture

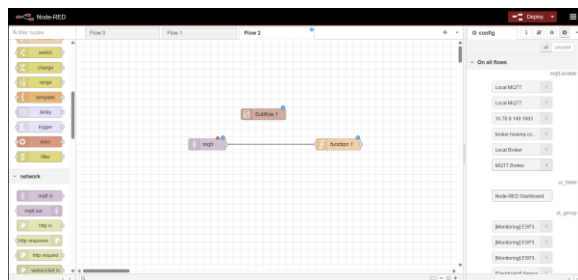
Sensor *soil moisture* adalah perangkat yang digunakan untuk mengukur kadar kelembapan tanah, sehingga sangat ideal dalam berbagai aplikasi seperti pertanian, dan proyek IoT yang berkaitan dengan monitoring lingkungan [15].

Sensor ini bekerja dengan prinsip mendeteksi perubahan resistansi atau kapasitansi tanah, yang dipengaruhi oleh kandungan air di dalamnya. Sensor ini umumnya terdiri dari dua *probe* logam atau pelat kapasitif yang ditanam di tanah untuk mengukur tingkat kelembapan, kemudian menghasilkan data berupa sinyal analog atau digital yang dapat dibaca oleh mikrokontroler.

2.7. Node-RED

Node-RED merupakan sebuah perangkat lunak berbasis *flow-based programming* yang dikembangkan oleh IBM dan dirancang untuk memudahkan integrasi perangkat keras, API, dan layanan daring dalam suatu IoT. *Node-RED* menggunakan antarmuka berbasis web yang intuitif, memungkinkan pengguna untuk membuat alur logika atau *flow* hanya dengan menyusun blok-blok visual (*nodes*) yang saling terhubung, tanpa memerlukan pemrograman tingkat lanjut [16].

Dalam konteks komunikasi data, *Node-RED* mendukung berbagai protokol, termasuk MQTT, yang banyak digunakan untuk pengiriman data sensor secara ringan dan efisien. Ilustrasi dari *Node-RED* dapat dilihat pada Gambar 3.



Gambar 3. Dashboard Node-RED

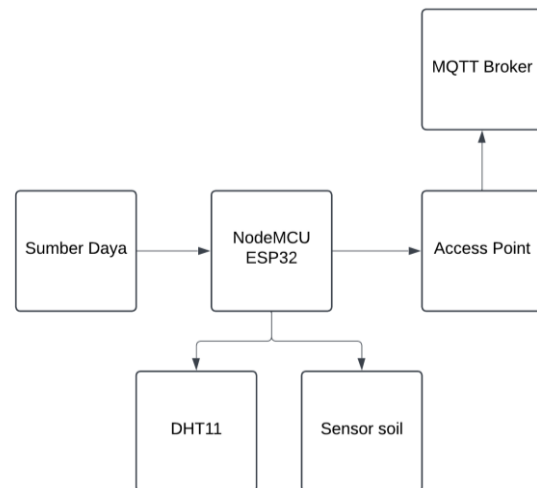
3. METODE PENELITIAN

Metode penelitian menjelaskan tahapan digunakan dalam penelitian ini. Bagian ini mencakup cara proses perancangan perangkat keras dan lunak, serta pengujian sistem yang telah dibangun untuk memperoleh data yang diperlukan.

3.1. Perancangan Perangkat Keras

Perangkat keras pada penelitian ini dirancang untuk memenuhi kebutuhan sistem monitoring berbasis IoT yang efisien dari segi energi maupun biaya. Komponen inti yang digunakan adalah modul ESP32, yakni sebuah mikrokontroler terintegrasi dengan kemampuan komunikasi nirkabel Wi-Fi. Sistem ini mengintegrasikan dua jenis sensor, yaitu sensor DHT11 untuk pemantauan suhu dan kelembapan udara, serta sensor kelembapan tanah untuk mendeteksi kadar air dalam media tanam. Masing-masing sensor memiliki karakteristik konsumsi daya yang berbeda.

Node ini dirancang mampu membaca data kelembapan tanah, suhu, dan kelembapan udara, kemudian mengirimkan data tersebut ke MQTT broker secara periodik melalui jaringan WiFi. *Soil moisture* Sensor dihubungkan ke pin ADC ESP32. Nilai ADC diubah menjadi persentase kelembapan. DHT11 dihubungkan ke pin digital untuk pembacaan data suhu dan kelembapan udara. Sumber daya menggunakan adaptor 5V. Rancangan blok dapat dilihat pada Gambar 4.



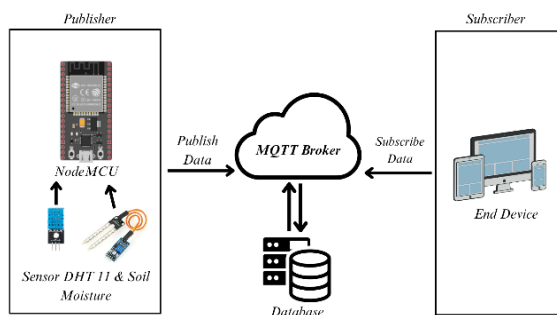
Gambar 4. Diagram Blok Perangkat Keras

3.2. Perancangan Perangkat lunak

Perancangan algoritma FSM yang dilakukan pada *NodeMCU* ESP32 dimulai Pada *state Disconnected*, perangkat akan memeriksa kembali status koneksi WiFi dan mencoba menyambungkan ulang jika terputus. Bila koneksi WiFi berhasil, maka sistem akan masuk ke *state Connecting* untuk mencoba menginisialisasi koneksi dengan broker MQTT. Jika proses koneksi berhasil, perangkat berpindah ke *state Connected* yang merupakan keadaan siaga untuk melakukan pengiriman data.

Secara berkala, yaitu setiap 20 detik, sistem akan berpindah ke *state Publishing* untuk melakukan simulasi pembacaan data sensor seperti temperatur, kelembapan udara, dan kelembapan tanah dengan nilai, kemudian mengemas data tersebut dalam format JSON dan mengirimkannya ke topik MQTT tertentu.

Seluruh proses komunikasi dan pengelolaan *state* dikendalikan melalui fungsi loop yang akan secara terus-menerus memantau dan mengeksekusi tindakan sesuai dengan kondisi sistem. Pendekatan FSM ini memungkinkan sistem menjadi lebih tangguh, modular, dan efisien dalam menangani perubahan koneksi serta menjamin kontinuitas pengiriman data secara berkala dalam jaringan IoT. Perancangan pada protokol MQTT dilakukan untuk memastikan proses komunikasi data antar *node* sensor dan *server* berjalan. Dalam implementasinya, setiap *NodeMCU* ESP32 dikonfigurasi sebagai *publisher* yang secara periodik mengirimkan data sensor seperti suhu, kelembapan udara, dan kelembapan tanah ke *broker* MQTT melalui topik sensor/data sebagai bentuk evaluasi performa transmisi. Protokol ini dirancang agar dapat beroperasi. Rancangan Arsitektur Protokol MQTT dapat dilihat pada Gambar 5.



Gambar 5. Rancangan Arsitektur Protokol MQTT

3.3. Perancangan Pengujian

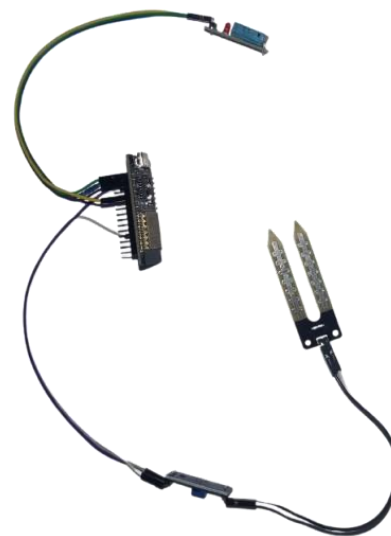
Pengujian dirancang untuk mengukur konsumsi energi *node* IoT secara sistematis melalui pencatatan durasi operasional pada setiap *state* FSM. Satu siklus pengujian berlangsung selama 3.600 detik atau setara dengan 1 jam, di mana *node* akan melewati seluruh *state* FSM secara berurutan. Pengujian secara keseluruhan dilaksanakan dalam dua sesi terpisah, di mana setiap sesi terdiri atas 6 siklus operasional sehingga setiap sesi berlangsung selama 6 jam. Pemisahan sesi pengujian ini bertujuan untuk mengamati konsistensi dan keterulangan profil konsumsi energi *node* pada kondisi waktu operasional yang berbeda. Dengan demikian, total keseluruhan pengujian mencakup siklus 12 jam yang terbagi ke dalam dua sesi.

4. HASIL DAN PEMBAHASAN

Hasil-hasil yang didapatkan dari tahapan implementasi yang dilakukan, Berikut adalah hasil implementasi yang diperoleh dari penelitian ini.

4.1. Implementasi Perangkat Keras

Implementasi perangkat keras melibatkan penerapan komponen-komponen yang dibutuhkan oleh sistem yang telah dirancang. Implementasi sistem dilakukan menggunakan *node* yang dilengkapi dengan sensor *Soil moisture* dan DHT11 Temperature & Humidity. Sensor *Soil moisture* dihubungkan ke pin ADC GPIO34 untuk membaca nilai kelembapan tanah dalam bentuk sinyal analog yang kemudian diubah menjadi persentase kelembapan menggunakan fungsi mapping. Sensor DHT11 dihubungkan ke pin digital GPIO4 untuk mengukur suhu udara (°C) dan kelembapan udara (%). Adapun Gambaran dari hasil implementasi dapat dilihat pada Gambar 6.



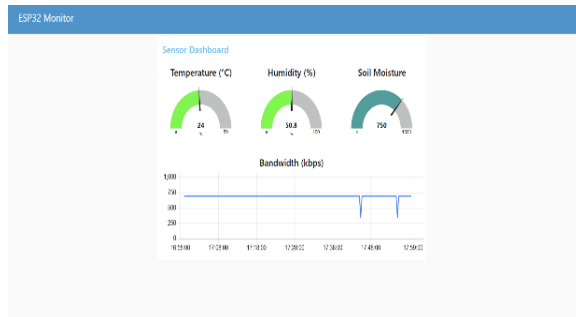
Gambar 6. Implementasi Perangkat Keras

4.2. Implementasi Perangkat Lunak

Penerapan sistem komunikasi MQTT pada ESP32 dalam penelitian ini dilakukan dengan menggunakan mikrokontroler ESP32 sebagai *node* pengirim data yang terhubung dengan sensor *soil moisture* dan DHT11. Data hasil pembacaan sensor kemudian dikirim ke *broker* MQTT melalui jaringan WiFi. Protokol MQTT digunakan karena mendukung *publish-subscribe pattern*, yang meminimalkan *overhead* komunikasi dan sangat cocok untuk transmisi data periodik dari *node* sensor ke *server* pusat. Dalam implementasinya, ESP32 berhasil melakukan koneksi ke *broker* MQTT publik dan secara berkala mengirimkan data dalam format JSON ke topik sensor/data. Data tersebut kemudian dapat diterima oleh *subscriber* atau ditampilkan dalam *dashboard*. Hasil dari pengiriman data ini juga terverifikasi melalui *Serial Monitor*. Data dari sensor berhasil dikirim setiap siklus tertentu tanpa kehilangan paket yang signifikan. Jika terjadi kehilangan koneksi, ESP32 dapat kembali masuk ke *state Connecting* dan mencoba menghubungkan ulang ke *broker* MQTT.

Dari sisi *node* sensor mengirimkan datanya langsung ke *broker* pusat. Ini memudahkan pemantauan dan sinkronisasi data, serta memungkinkan manajemen daya per-*node* secara

individual. Secara keseluruhan, hasil implementasi menunjukkan bahwa sistem komunikasi MQTT pada ESP32 dapat berjalan efektif dan efisien dalam skenario pemantauan berbasis sensor. Adapun Gambaran dari *dashboard node* pemantauan dapat dilihat pada Gambar 7.



Gambar 7. Tampilan Dashboard Pemantauan

4.3. Pengujian Sistem

Pengujian *node IoT* dilaksanakan dengan merekam log durasi operasional pada setiap *state* FSM selama satu siklus penuh yang berlangsung selama 3.600 detik atau setara dengan 1 jam. Berdasarkan data log yang diperoleh, *node* memulai siklus dari *State Disconnected* selama 3 detik, yaitu kondisi awal saat *node* belum terhubung ke jaringan maupun *broker* MQTT. Selanjutnya, *node* bertransisi ke *State Connecting* selama 7 detik, di mana mikrokontroler melakukan proses koneksi Wi-Fi sekaligus *handshake* protokol MQTT dengan *broker* yang telah dikonfigurasi. Tabel pengukuran waktu *state* dapat dilihat pada

Tabel 1. Log State pada Siklus Satu Jam

State	Durasi (s)	Porsi waktu (%)	Kontribusi Energi (Wh)
Disconnected	3	0,08	0,0003
Connecting	7	0,19	0,0008
Connected	3.375	93,75	0,4088
Publishing	215	5,98	0,0261
Total	3.600	100	0,4360

Setelah koneksi berhasil terjalin, *node* memasuki *State Connected* dengan durasi terpanjang dalam siklus ini, yakni selama 3.375 detik. Pada *state* ini, *node* berada dalam kondisi terhubung dan aktif memantau kondisi sensor sambil menunggu interval waktu yang ditentukan sebelum melakukan publikasi data. *State* ini mendominasi keseluruhan siklus dengan proporsi sebesar 93,75% dari total waktu operasional, yang mencerminkan bahwa *node* menghabiskan sebagian besar waktunya dalam kondisi siaga sambil menjaga koneksi tetap aktif ke *broker* MQTT. Terakhir, *node* bertransisi ke *State Publishing* selama 215 detik, yaitu proses pengiriman data hasil pembacaan sensor *Soil moisture* dan DHT11 ke *broker* MQTT. Durasi *state* ini mencakup 5,98% dari total siklus, mencakup seluruh rangkaian aktivitas publish

mulai dari pembacaan nilai sensor, pembentukan paket data, hingga konfirmasi pengiriman dari *broker*. Berdasarkan pengujian yang telah dilaksanakan selama satu siklus penuh operasional *node IoT* selama 3.600 detik (1 jam), total konsumsi energi yang terukur pada implementasi FSM adalah sebesar 0,4360 Wh. Nilai ini merupakan akumulasi dari konsumsi energi pada seluruh *state* FSM.

Pengujian konsumsi energi *node IoT* dengan implementasi FSM dilaksanakan dalam 2 sesi pengujian terpisah, di mana setiap sesi terdiri atas 6 siklus operasional dengan durasi masing-masing siklus selama 3.600 detik (1 jam). Dengan demikian, setiap sesi berlangsung selama 6 jam, dan total keseluruhan pengujian mencakup 12 siklus dalam 12 jam yang terbagi ke dalam dua sesi yang dilaksanakan secara terpisah. Berikut adalah tabel pengukuran selama 2 sesi dapat dilihat pada

Tabel 2. Data Pengujian Selama 2 Sesi

Sesi	Siklus (jam)	Konsumsi Energi (Wh)
1	1	0,4360
	2	0,4160
	3	0,4070
	4	0,4940
	5	0,4020
	6	0,4120
Total Sesi 1		2,5670
2	7	0,4400
	8	0,4370
	9	0,4350
	10	0,4290
	11	0,4250
	12	0,4320
Total Sesi 2		2,5980
Total		5,1650

Dari 12 data pengukuran yang diperoleh, dilakukan analisis statistik dasar untuk memahami karakteristik distribusi konsumsi energi *node IoT* selama pengujian berlangsung, Tabel statistik dapat dilihat pada

Tabel 3. Parameter Statistik Pengukuran

Parameter Statistik	Sesi 1 (Wh)	Sesi 2 (Wh)
Nilai Tertinggi	0,4940	0,4400
Nilai Terendah	0,4020	0,4250
Total Energi	2,5670	2,5980
Rata-rata per Siklus	0,4278	0,4330
Range (Maks–Min)	0,0920	0,0150

Pengujian dilaksanakan dalam dua sesi terpisah, masing-masing berlangsung selama 6 jam dengan 6 siklus pengukuran per sesi. Pemisahan sesi pengujian ini bertujuan untuk mengamati konsistensi perilaku konsumsi energi *node IoT* pada kondisi pengoperasian yang berbeda waktu, sekaligus memverifikasi bahwa implementasi FSM menghasilkan profil daya yang berulang dan dapat diandalkan lintas sesi. Pada Sesi 1,

total konsumsi energi yang terukur selama 6 jam operasional adalah sebesar 2,5670 Wh dengan nilai rata-rata per siklus 0,4278 Wh. Data pada sesi ini menunjukkan variasi yang cukup mencolok, terutama pada siklus ke-4 yang mencatat nilai tertinggi sebesar 0,4940 Wh dan siklus ke-5 yang langsung turun ke nilai terendah 0,4020 Wh. Fluktuasi tajam antara kedua siklus yang berurutan ini mengindikasikan adanya gangguan sementara pada kondisi jaringan di sekitar siklus ke-4, kemungkinan besar berupa penurunan kualitas sinyal Wi-Fi yang memaksa *node* melakukan proses retransmission paket MQTT atau perpanjangan durasi *State Connecting*, sehingga konsumsi energi melonjak secara tidak normal. Kemudian pada siklus berikutnya, kondisi jaringan kembali stabil dan konsumsi energi turun ke titik terendah dalam keseluruhan pengujian. Selain anomali pada siklus ke-4, siklus-siklus lain pada Sesi 1 berada pada rentang 0,4070 Wh hingga 0,4360 Wh yang mencerminkan operasional FSM yang cukup stabil.

Pada Sesi 2, total konsumsi energi yang terukur selama 6 jam operasional adalah sebesar 2,5980 Wh dengan nilai rata-rata per siklus 0,4330 Wh. Berbeda dengan Sesi 1, data pada sesi ini memperlihatkan konsistensi yang jauh lebih baik, di mana seluruh 6 siklus berada dalam rentang sempit 0,4250 Wh hingga 0,4400 Wh dengan range hanya sebesar 0,0150 Wh. Kestabilan data pada Sesi 2 ini mengindikasikan bahwa kondisi jaringan dan lingkungan operasional pada sesi tersebut berada dalam keadaan ideal tanpa gangguan eksternal yang berarti, sehingga FSM dapat menjalankan seluruh transisi *state* secara konsisten dari satu siklus ke siklus berikutnya. Apabila kedua sesi dibandingkan, terdapat selisih total konsumsi energi sebesar 0,0310 Wh antara Sesi 1 dan Sesi 2, sementara selisih rata-rata per siklusnya hanya sebesar 0,0052 Wh atau kurang dari 1,2%. Selisih yang sangat kecil ini menunjukkan bahwa secara keseluruhan, kedua sesi menghasilkan profil konsumsi energi yang setara dan konsisten, dan perbedaan yang ada lebih disebabkan oleh anomali pada siklus ke-4 di Sesi 1 daripada perbedaan perilaku sistematis antara kedua sesi. Hal ini memperkuat kesimpulan bahwa implementasi FSM pada *node* IoT menghasilkan konsumsi energi yang stabil, berulang, dan tidak bergantung pada waktu pelaksanaan sesi pengujian.

Secara keseluruhan, dari total 12 siklus yang telah diuji, 11 siklus menghasilkan nilai konsumsi energi dalam rentang 0,4020 Wh hingga 0,4400 Wh, sementara hanya 1 siklus yang berada di luar rentang tersebut akibat kondisi jaringan yang tidak stabil. Dengan nilai rata-rata keseluruhan sebesar 0,4304 Wh per siklus, total konsumsi energi *node* IoT ini diproyeksikan sebesar $\pm 10,33$ Wh per hari apabila dioperasikan secara kontinu selama 24 jam, yang menjadi acuan teknis penting dalam perancangan kapasitas sumber daya untuk deployment jangka panjang di lapangan.

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil menganalisis konsumsi energi ESP32 yang terintegrasi dengan sensor *Soil moisture* dan DHT11 menggunakan protokol MQTT berbasis pendekatan FSM. Berdasarkan pengujian selama 12 siklus operasional yang terbagi dalam dua sesi, diperoleh total konsumsi energi rata-rata per siklus sebesar 0,4304 Wh dengan proyeksi konsumsi harian sebesar $\pm 10,33$ Wh apabila dioperasikan secara kontinu selama 24 jam. Dari keseluruhan siklus operasional, *state Connected* terbukti mendominasi waktu operasional sebesar 93,75% dari total siklus dan menjadi kontributor terbesar konsumsi energi sistem, diikuti oleh *state Publishing* sebesar 5,98%, *Connecting* sebesar 0,19%, dan *Disconnected* sebesar 0,08%. Implementasi FSM juga terbukti menghasilkan profil konsumsi energi yang stabil dan berulang lintas sesi pengujian, dengan selisih rata-rata antar sesi hanya sebesar 0,0052 Wh atau kurang dari 1,2%. Anomali konsumsi energi yang terjadi pada siklus ke-4 Sesi 1 disebabkan oleh gangguan kualitas sinyal Wi-Fi yang memicu retransmisi paket MQTT, bukan oleh kelemahan sistematis pada implementasi FSM. Secara keseluruhan, sistem komunikasi MQTT pada ESP32 mampu berjalan efektif dan andal dalam skenario pemantauan berbasis sensor, dengan kemampuan pemulihan koneksi otomatis melalui mekanisme transisi *state* FSM yang terstruktur.

DAFTAR PUSTAKA

- [1] Abdullah, M. Zainul Haq, Cholish, M. Putri, and A. Ramadhan, "Implementasi Internet Of Things Dalam Pemantauan Optimal Kerja Panel Surya," *Rekayasa Elektrikal dan Energi*, vol. 4, no. 2, pp. 152–157, Jan. 2015, doi: 10.30596/relev.v1i1.
- [2] F. Tsvetanov and I. Georgieva, "Modeling of Energy Consumption of Sensor Nodes," in *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*, 2020, pp. 431–436. doi: 10.23919/MIPRO48935.2020.9245445.
- [3] M. Rana, Q. Mamun, and R. Islam, "Balancing Security and Efficiency: A Power Consumption Analysis of a Lightweight Block Cipher," *Electronics (Switzerland)*, vol. 13, no. 21, Nov. 2024, doi: 10.3390/electronics13214325.
- [4] I. Narita, F. Popescu, and C. Despina-Stoian, "Internet of Things Communication Systems based on MQTT and BLE Protocols," *Journal of Military Technology*, vol. 4, no. 2, pp. 39–44, Dec. 2021, doi: 10.32754/JMT.2021.2.08.
- [5] H. J. Jara Ochoa, R. Peña, Y. Ledo Mezquita, E. Gonzalez, and S. Camacho-Leon, "Comparative Analysis of Power Consumption between MQTT and HTTP Protocols in an IoT Platform Designed and Implemented for Remote Real-Time Monitoring of Long-Term Cold Chain Transport Operations," *Sensors*, vol. 23, no. 10, May 2023, doi: 10.3390/s23104896.

- [6] T. Thevananthan and V. J. Vahini, "Smart, Easy End-User Installation and Reliable Node-Coordinator System & Protocol solution for IoT Applications using Cost-Effective Embedded systems," in *Proceedings of the 112th Annual Sessions of the Institution of Engineers, Sri Lanka (IESL)*, Institution of Engineers, Sri Lanka (IESL), 2018.
- [7] F. Aziz, F. Said, and A. Sudrajat, "Penerapan Konsep Finite State Automata Dalam Proses Pendaftaran Kelas Kursus Bahasa Inggris Pada Tempat Kursus," *MATICS*, vol. 12, no. 2, pp. 93–98, Mar. 2021, doi: 10.18860/mat.v12i2.9330.
- [8] O. O. Akintade, T. K. Yesufu, and L. O. Kehinde, "Development of Power Consumption Models for ESP8266-Enabled Low-Cost IoT Monitoring Nodes," *Advances in Internet of Things*, vol. 09, no. 01, pp. 1–14, 2019, doi: 10.4236/ait.2019.91001.
- [9] S. Jeddou, F. Fernández, L. Diez, A. Baina, N. Abdallah, and R. Agüero, "Delay and Energy Consumption of MQTT over QUIC: An Empirical Characterization Using Commercial-Off-The-Shelf Devices," *Sensors*, vol. 22, no. 10, May 2022, doi: 10.3390/s22103694.
- [10] C. Bordin, A. Håkansson, and S. Mishra, "Smart Energy and power systems modelling: An IoT and Cyber-Physical Systems perspective, in the context of Energy Informatics," in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 2254–2263. doi: 10.1016/j.procs.2020.09.275.
- [11] A. R. Utamy, Siswanto, and Sutarti, "Prototype Wireless Sensor Network Sistem Pengukuran Debu Dan Suhu Udara Berbasis Mqtt Server," vol. 10, no. 2, 2023, [Online]. Available: <https://www.amazon.in/REES52-Traffic-Light->
- [12] M. G. Khairi, "Penerapan Finite State Machine Dalam Game Pada Model Karakter Pemain Menggunakan Godot Engine The Application Of Finite State Machine In Games For Playable Character Model Using Godot Engine," 2024.
- [13] A. Anggrawan, S. Hadi, and C. Satria, "IoT-Based Garbage Container System Using NodeMCU ESP32 Microcontroller," *Journal of Advances in Information Technology*, vol. 13, no. 6, pp. 569–577, Dec. 2022, doi: 10.12720/jait.13.6.569-577.
- [14] F. Ztt, E. Mashonjowa, and E. Matandirotya, "DHT11 Based Temperature and Humidity Measuring System," *Journal of Electrical Engineering and Electronic Technology*, vol. 11, no. 5, 2022, doi: 10.4172/jeet.1000902.
- [15] N. Effendi, W. Ramadhani, and F. Farida, "Perancangan Sistem Penyiraman Tanaman Otomatis Menggunakan Sensor Kelembapan Tanah Berbasis IoT," *Jurnal CoSciTech (Computer Science and Information Technology)*, vol. 3, no. 2, pp. 91–98, Aug. 2022, doi: 10.37859/coscitech.v3i2.3923.
- [16] K. Ferencz and J. Domokos, "Using Node-RED platform in an industrial environment," 2020. [Online]. Available: <https://www.researchgate.net/publication/339596157>