SKRIPSI

IMPLEMENTASI ALGORITMA *DATA ENCRYPTION STANDARD* (DES) DAN TEKNIK *GRAY CODE* PADA PROTOKOL *MESSAGE QUEUING TELEMETRY TRANSPORT* (MQTT)

IMPLEMENTATION OF DATA ENCRYPTION STANDARD (DES) ALGORITHM AND GRAY CODE TECHNIQUE IN MESSAGE QUEUING TELEMETRY TRANSPORT (MQTT) PROTOCOL

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

HENDRA USMAN
D0221079

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS SULAWESI BARAT
MAJENE
2025

SKRIPSI

IMPLEMENTASI ALGORITMA *DATA ENCRYPTION STANDARD* (DES) DAN TEKNIK *GRAY CODE* PADA PROTOKOL *MESSAGE QUEUING TELEMETRY TRANSPORT* (MQTT)

HENDRA USMAN
D0221079

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS SULAWESI BARAT
MAJENE
2025

LEMBAR PERSETUJUAN

SKRIPSI

IMPLEMENTASI ALGORITMA *DATA ENCRYPTION STANDARD* (DES) DAN TEKNIK *GRAY CODE* PADA PROTOKOL *MESSAGE QUEUING TELEMETRY TRANSPORT* (MQTT)

Telah dipersiapkan dan disusun oleh

HENDRA USMAN
NIM. D0221079

Telah dipertahankan di depan Tim Penguji

Pada tanggal 7 April 2025

Susunan Tim Penguji

Pembimbing I

Dr. Ir. Adam M Tanniewa, S.Kom., S.E., M.M., M.T., IPM., ASEAN. Eng.
NIDN. 0915057702

Penguji I

Muh. Rafli Rasyid, S.Kom., M.T.
NIP. 198808182022031006

Pembimbing II

A. Amirul Asnan Cirua, S.T., M.Kom.
NIP. 199804022024061001

Penguji II

Siti Aulia Rachmini, S.T., M.T.
NIP. 198207062008042003

Penguji III

Wawan Firgiawan, S.T., M.Kom.
NIDK. 8948080023

LEMBAR PENGESAHAN

IMPLEMENTASI ALGORITMA *DATA ENCRYPTION STANDARD* (DES) DAN TEKNIK *GRAY CODE* PADA PROTOKOL *MESSAGE QUEUING TELEMETRY TRANSPORT* (MQTT)

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer
Disusun oleh:

HENDRA USMAN
NIM. D0221079

Skripsi ini telah diuji dan dinyatakan lulus
pada tanggal 7 Mei 2025
Telah diperiksa dan disetujui oleh:

Pembimbing I

Dr. Ir. Adam M Tanniewa, S.Kom., S.E., M.M., M.T., IPM., ASEAN. Eng.
NIDN. 0915057702

Pembimbing II

A. Amirul Asnan Cirua, S.T., M.Kom.
NIP. 199804022024061001

Dekan Fakultas Teknik

Prof. Dr. Ir. Hafsah Nirwana, M.T.
NIP. 196404051990032002

Ketua Program Studi Informatika

Muh. Ilhli Rasyid, S.Kom., M.T.
NIP. 198808182022031006

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku **(UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).**

Majene, 7 Mei 2025

Hendra Usman

NIM: D0221079

ABSTRAK

Hendra Usman. Implementasi Algoritma *Data Encryption Standard* (DES) dan Teknik *Gray Code* pada Protokol *Message Queuing Telemetry Transport* (MQTT). (Dibimbing oleh **Adam M Tanniewa** dan **A. Amirul Asnan Cirua**).

Sistem komunikasi pada *Internet of Things* (IoT) banyak memanfaatkan protokol *Message Queuing Telemetry Transport* (MQTT) karena efisiensinya dalam pengiriman data pada jaringan dengan keterbatasan sumber daya. Namun, secara default, MQTT tidak menyediakan mekanisme enkripsi yang memadai sehingga rentan terhadap penyadapan dan pencurian data. Untuk mengatasi permasalahan tersebut, penelitian ini mengimplementasikan algoritma *Data Encryption Standard* (DES) yang dikombinasikan dengan teknik *Gray Code* guna meningkatkan keamanan proses enkripsi dan dekripsi data pada protokol MQTT. Sistem dikembangkan dengan mengintegrasikan sensor DHT22 dan mikrokontroler ESP8266 serta ESP32, dilengkapi dengan dua *library* khusus yaitu *"GrayCodeToDES"* untuk enkripsi dan konversi *Gray Code*, serta *"DESGrayDecoder"* untuk dekripsi dan *Inverse Gray Code*. Hasil pengujian menunjukkan bahwa sistem mampu mengirimkan dan menerima data dengan akurasi 100% serta dapat mendeteksi kesalahan koneksi sensor dan *broker* secara efektif. Dari sisi performa, ESP32 menunjukkan kinerja yang lebih stabil dan responsif dibandingkan ESP8266, dengan waktu *delay*, enkripsi, dan dekripsi yang lebih cepat, serta penggunaan memori yang lebih efisien. Analisis keamanan melalui pengujian *entropi ciphertext* menghasilkan nilai mendekati batas maksimal untuk 64-bit, menunjukkan tingkat keacakan data yang tinggi. Selain itu, *Character Error Rate* (CER) tercatat 0% di seluruh skenario pengujian, membuktikan transmisi data berjalan tanpa kesalahan karakter. Berdasarkan hasil tersebut, kombinasi algoritma DES dan teknik *Gray Code* terbukti efektif dalam meningkatkan keamanan, kecepatan, dan keandalan komunikasi data berbasis MQTT pada sistem IoT. ESP32 direkomendasikan untuk aplikasi berskala besar yang memerlukan kestabilan dan efisiensi jangka panjang, sementara ESP8266 lebih cocok untuk aplikasi IoT dengan kebutuhan sumber daya yang lebih ringan.

Kata Kunci: MQTT, IoT, *Data Encryption Standard* (DES), *Gray Code*, Keamanan Data, ESP8266, ESP32.

ABSTRACT

Hendra Usman. *Implementation of the Data Encryption Standard (DES) Algorithm and Gray Code Technique on the Message Queuing Telemetry Transport (MQTT) Protocol. (Supervised by* ***Adam M. Tanniewa*** *and* ***A. Amirul Asnan Cirua****)*

Communication systems in the Internet of Things (IoT) widely utilize the Message Queuing Telemetry Transport (MQTT) protocol due to its efficiency in data transmission over networks with limited resources. However, by default, MQTT does not provide adequate encryption mechanisms, making it vulnerable to eavesdropping and data theft. To address this issue, this study implements the Data Encryption Standard (DES) algorithm combined with the Gray Code technique to enhance the security of the encryption and decryption processes on the MQTT protocol. The system was developed by integrating the DHT22 sensor and microcontrollers ESP8266 and ESP32, equipped with two specialized libraries: "GrayCodeToDES" for encryption and Gray Code conversion, and "DESGrayDecoder" for decryption and Gray Code inversion. The testing results show that the system is capable of transmitting and receiving data with 100% accuracy and can effectively detect sensor and broker connection errors. In terms of performance, the ESP32 demonstrates more stable and responsive operation compared to the ESP8266, with lower transmission delay, encryption, and decryption times, along with more efficient memory usage. Security analysis through ciphertext entropy testing yielded values approaching the maximum limit for 64-bit data, indicating a high level of data randomness. Additionally, the Character Error Rate (CER) was recorded at 0% across all testing scenarios, proving that data transmission occurred without any character errors. Based on these results, the combination of the DES algorithm and the Gray Code technique is proven to be effective in enhancing the security, speed, and reliability of MQTT-based data communication systems in IoT applications. ESP32 is recommended for large-scale applications requiring long-term stability and efficiency, while ESP8266 is better suited for lightweight IoT applications with limited resource requirements.

Keywords: MQTT, IoT, Data Encryption Standard (DES), Gray Code, Data Security, ESP8266, ESP32.

BAB I
PENDAHULUAN

1.1. Latar Belakang

Protokol *Message Queuing Telemetry Transport* (MQTT) adalah salah satu protokol yang banyak diterapkan dalam komunikasi di *Internet of Things* (IoT) (Alfasa, Dewanta and Istikmal, 2024). MQTT diperkenalkan oleh *Organization for the Advancement of Structured Information Standards* (OASIS) pada tahun 2013. Protokol ini memfasilitasi komunikasi data *Machine-to-Machine* (M2M) pada tingkat aplikasi, memungkinkan perangkat-perangkat IoT untuk berinteraksi secara efisien (Mishra and Kertesz, 2020). MQTT dirancang untuk perangkat dengan sumber daya terbatas dan jaringan yang memiliki *bandwidth* rendah, latensi tinggi, serta perangkat dengan keandalan yang kurang (Diono *et al.*, 2021). Protokol MQTT menggunakan model komunikasi *publish-subscribe* yang melibatkan tiga komponen utama: *publisher*, *subscriber*, dan *message broker*. *Publisher* bertindak sebagai pengirim data, seperti data sensor, sementara *subscriber* berfungsi sebagai penerima yang mendaftar untuk menerima data tersebut. *Message broker*, sebagai penghubung, mengelola dan menyampaikan data dari *publisher* ke *subscriber*. Dengan cara ini, *publisher* dan *subscriber* tidak perlu berinteraksi secara langsung, karena *message broker* menangani distribusi pesan berdasarkan langganan yang ada (Pratama, Wicaksono and Pramudhita, 2023).

Meskipun efisien, protokol ini memiliki beberapa kelemahan yang signifikan. Dalam kondisi *default*, protokol ini tidak menyediakan metode keamanan yang memadai untuk melindungi privasi data dan integritas data. Salah satunya adalah kurangnya mekanisme keamanan bawaan dalam proses pengiriman data. Selama pengiriman data, MQTT sangat rentan terhadap penyadapan dan pencurian data, karena tidak dilengkapi enkripsi secara *default*. Hal ini membuat komunikasi menjadi rentan terhadap serangan yang dapat mengakibatkan pengungkapan informasi sensitif (Akbar, Bahri and Nirmala, 2024). Untuk memperkuat kerahasiaan (*confidentiality*) data yang dikirimkan melalui MQTT, diperlukan penambahan algoritma kriptografi untuk mengenkripsi data sebelum dikirimkan. Enkripsi (*encipher*) adalah proses mengubah pesan atau data yang berupa *plaintext*

(pesan yang dapat dimengerti oleh manusia) menjadi *ciphertext* (pesan acak dan sulit dimengerti oleh manusia). Sebaliknya, dekripsi (*decipher*) mengembalikan *ciphertext* ke bentuk semula, yaitu *plaintext* (Alfasa, Dewanta and Istikmal, 2024). Kriptografi bertujuan untuk menjaga kerahasiaan pesan, memastikan integritas data, melakukan otentikasi pengirim dan penerima, serta mencegah penyangkalan pengiriman atau penerimaan pesan (Hidayatulloh *et al.*, 2023). Meskipun terjadi serangan dan mengakibatkan pesan berhasil dicuri, isi dari pesan tersebut tetap terlindungi karena sudah terenkripsi dan akan sulit dimengerti oleh pihak yang tidak berwenang (Apinento, Abdan and Riel, 2023).

Salah satu algoritma yang dapat diterapkan adalah *Data Encryption Standard* (DES). *National Institute of Standards and Technology* (NIST) mengenalkan algoritma ini sebagai standar kriptografi di tahun 1977(Pratama *et al.*, 2023). DES merupakan algoritma kriptografi simetris yang digunakan untuk mengenkripsi dan mendekripsi data. Algoritma ini menggunakan satu kunci yang sama untuk proses enkripsi dan dekripsi (Mahesti, Ciptaningtyas and Astungkara, 2025). Salah satu kelebihan DES yaitu lebih baik daripada algoritma XOR lainnya (Thahara and Siregar, 2021). Proses enkripsi pada DES terdiri dari beberapa tahapan penting, antara lain pembangkitan kunci (*key expansion*), pengulangan putaran (*rounds*), serta substitusi dan permutasi menggunakan *S-Box* dan *P-Box* (Pratama *et al.*, 2023). DES bekerja dengan mengubah 64-bit *plaintext* menjadi 64-bit *ciphertext* menggunakan kunci internal sepanjang 56-bit (juga disebut *subkey*). Kunci internal ini dihasilkan dari kunci eksternal yang memiliki panjang 64-bit. Kunci utama DES sebenarnya berukuran 64-bit, namun hanya 56-bit yang digunakan untuk enkripsi karena setiap bit kedelapan adalah bit paritas yang dibuang (Al-hazaimeh *et al.*, 2023). Enkripsi dilakukan melalui 16 putaran pada blok data sebelum akhirnya menghasilkan blok data yang sudah terenkripsi (Pratama *et al.*, 2023).

Meskipun demikian, kunci efektif sepanjang 56-bit pada DES membuatnya kurang ideal dalam menghadapi serangan keamanan dengan kekuatan komputasi modern (Al-hazaimeh *et al.*, 2023). Oleh karena adanya kelemahan tersebut, diperlukan penambahan teknik tertentu untuk meningkatkan keamanan DES. Salah satu metode yang dapat digunakan untuk mengoptimalkan algoritma adalah dengan memanfaatkan *Gray Code. Gray Code* adalah suatu urutan pengkodean di mana

dua kode yang berurutan hanya berbeda dalam satu posisi bit. Ini berarti bahwa ketika berpindah dari satu kode ke kode berikutnya, hanya satu bit yang berubah (Bouyuklieva *et al.*, 2024). Dengan sifat ini, penggabungan *Gray Code* dan DES berpeluang meningkatkan keamanan dalam proses enkripsi dan dekripsi dengan kompleksitas tambahan yang dihasilkannya sehingga algoritma menjadi lebih sulit untuk diprediksi dibandingkan dengan versi asli DES.

Beberapa penelitian telah mengimplementasikan algoritma kriptografi ke dalam protokol MQTT. Salah satunya oleh Akbar, Bahri and Nirmala (2024). Penelitian ini menunjukkan bahwa algoritma RSA (*Rivest-Shamir-Adleman*) efektif dalam menjaga kerahasiaan, integritas, dan privasi data sensor IoT. Evaluasi menunjukkan peningkatan keamanan dengan rata-rata waktu enkripsi 0,34 ms dan dekripsi 0,576 ms per pesan yang membuktikan responsivitas dan keandalan algoritma ini.

Di sisi lain, beberapa penelitian juga telah mengembangkan metode tambahan untuk mengoptimalkan keamanan algoritma DES. Penelitian yang dilakukan oleh Ruziq, Sihombing and Sawaluddin (2020) menyimpulkan bahwa kombinasi algoritma DES dan *Lattice-Based Universal Cryptography* (LUC) dapat meningkatkan keamanan karena karena *cipher key* yang dihasilkan lebih panjang dan sulit ditebak. Selain itu, dalam penelitian yang dilakukan oleh Irawan and Winarno (2020) penggunaan gabungan algoritma DES dan AES memberikan lapisan keamanan tambahan yang kuat. Adapun Seif and Alexan (2020) mengusulkan skema keamanan berkapasitas tinggi berbasis *Gray Code*. Dalam penelitian ini, algoritma *Blowfish* digunakan untuk kriptografi, sementara *Gray Code* diterapkan pada Steganografi untuk mengacak urutan piksel dan menghindari bit *redundant*. Hasil pengujian menunjukkan skema ini efektif dalam meningkatkan kapasitas penyisipan data tanpa mengorbankan kualitas dan keamanan.

Berdasarkan uraian diatas dan didukung oleh penelitian sebelumnya, penelitian ini mengangkat judul **"Implementasi Algoritma *Data Encryption Standard* (DES) dan Teknik *Gray Code* pada Protokol *Message Queuing Telemetry Transport* (MQTT)"**. Dengan mengintegrasikan algoritma DES dan *Gray Code*, diharapkan dapat meningkatkan keamanan enkripsi dan dekripsi pada protokol MQTT.

1.2. Rumusan Masalah

Berdasarkan uraian latar belakang di atas, rumusan masalah dalam penelitian ini adalah bagaimana implementasi algoritma DES dan teknik *Gray Code* pada protokol MQTT?

1.3. Batasan Masalah

Adapun batasan masalah yang dapat dibuat berdasarkan latar belakang dan rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Penelitian ini menggunakan data yang diperoleh dari sistem sederhana. Data yang digunakan yaitu suhu dan kelembaban dari sensor DHT22.
2. Mikrokontroler yang digunakan dalam penelitian ini yaitu NodeMCU Amica ESP8266 dan ESP32 Devkit V1.
3. *Broker* yang digunakan yaitu mqtt.eclipseprojects.io
4. Algoritma yang digunakan dalam penelitian ini adalah DES dan teknik *Gray Code*.
5. Bahasa pemrograman yang digunakan dalam penelitian ini adalah C++ untuk *publisher* dan Python untuk *subscriber*.

1.4. Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah mengidentifikasi hasil implementasi algoritma DES dan teknik *Gray Code* pada protokol MQTT.

1.5. Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat besar dalam perkembangan teknologi, utamanya yaitu sebagai berikut:

1. Dapat mempelajari algoritma DES dan teknik *Gray Code* dalam protokol MQTT.
2. Memberikan kontribusi dalam pengembangan solusi enkripsi dan dekripsi yang lebih aman dalam protokol MQTT.

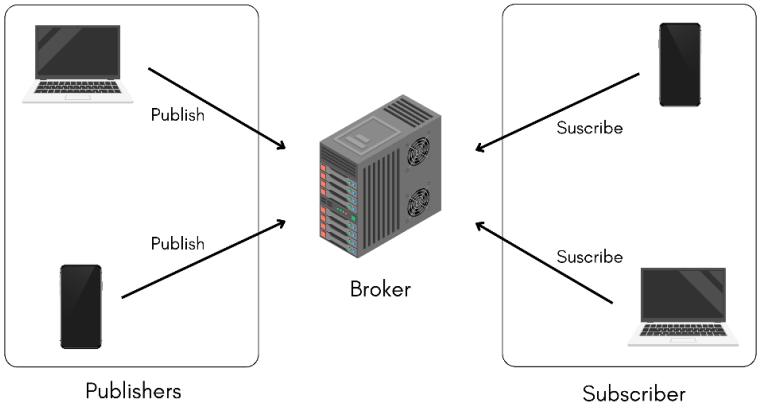
BAB II
TINJAUAN PUSTAKA

2.1. Landasan Teori

2.1.1. *Message Queuing Telemetry Transport* (MQTT)

Message Queuing Telemetry Transport (MQTT) adalah protokol komunikasi yang telah menjadi standar untuk IoT (Ahyadi *et al.*, 2021). MQTT dikenal sebagai protokol efektif untuk komunikasi *machine-to-machine* (M2M) dan aplikasi IoT, beroperasi pada lapisan aplikasi protokol TCP/IP (Hanif and Ilyas, 2024). MQTT dikembangkan oleh Andy Stanford-Clark dan Arlen Nipper pada tahun 1999. Kemudian, pada tahun 2013 OASIS (*Organization for the Advancement of Structured Information Standards*) memperkenalkan MQTT sebagai standar resmi dengan sertifikasi standar ISO/IEC 20922:2016 (Abduh, 2021).

Dalam komunikasi data berbasis model *client-server* tradisional, server hanya mengirimkan data saat ada permintaan dari *client*. Sebaliknya, dalam komunikasi yang menggunakan protokol MQTT, tidak ada hubungan *client-server*, melainkan menggunakan konsep *publisher* dan *subscriber* (Ahyadi *et al.*, 2021). Menurut Harnanta, Bhawiyuga and Basuki (2020), terdapat 3 (tiga) komponen utama dalam model komunikasi *publish-subscribe* meliputi *publisher*, *subscriber*, dan *broker* seperti yang disajikan dalam Gambar 2.1 berikut.



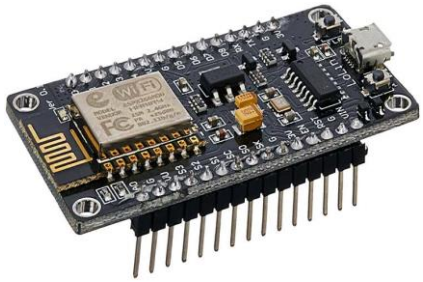
Gambar 2.1 Arsitektur protokol MQTT
(Sumber: Harnanta, Bhawiyuga and Basuki, 2020)

Publisher mengacu pada perangkat IoT yang secara berkala menerbitkan data yang dihasilkan oleh sensor, seperti pembacaan suhu, kelembaban, dan data lain yang relevan. Data ini disajikan dalam bentuk pesan, yang dapat diakses oleh pihak lain di jaringan (Handy and Pardede, 2022). Di sisi lain, *subscriber* adalah perangkat atau aplikasi yang mendaftar untuk menerima data yang dipublikasikan oleh *publisher* (Alshammari, 2023). Dengan demikian, *subscriber* dapat memperoleh informasi terbaru tanpa perlu secara aktif meminta data kepada *publisher*. Di tengah-tengah kedua komponen tersebut, *broker* memainkan peran yang sangat penting dalam arsitektur komunikasi ini. *Broker* bertanggung jawab untuk menyimpan dan mengelola data yang dipublikasikan oleh penerbit (Suprianto, Natasya and Riskiawan, 2023). Tugas *broker* tidak hanya terbatas pada penyimpanan, tetapi juga memastikan bahwa pesan dikirim ke pelanggan yang tepat berdasarkan topik yang diinginkan (Fikhri and Nurdin, 2024). Melalui mekanisme ini, model komunikasi *publish-subscribe* menciptakan jalur pertukaran informasi yang efisien dan terorganisir antara berbagai perangkat dalam jaringan.

Desain MQTT menekankan pada transfer data yang andal dengan keunggulan seperti kehilangan paket yang rendah, penggunaan *bandwidth* yang minimal, dan kebutuhan memori yang kecil, sehingga mudah beradaptasi di berbagai sektor (Hanif and Ilyas, 2024). Fleksibilitas MQTT terlihat jelas dalam berbagai aplikasinya, termasuk manajemen energi, perawatan kesehatan, telemetri, sistem jejaring sosial, otomatisasi rumah dan industri, pertanian pintar, transportasi, dan pemantauan lingkungan jarak jauh (Im and Lim, 2023).

2.1.2. NodeMCU ESP8266

NodeMCU ESP8266 merupakan *platform* IoT bersifat *open-source* yang menggunakan *System on Chip* (SoC) ESP8266. *Board* elektronik ini berfungsi ganda sebagai mikrokontroler sekaligus mendukung koneksi *Wireless Fidelity* (WiFi) (Manullang, Saragih and Hidayat, 2021). Gambar 2.2 menampilkan bentuk fisik dari NodeMCU.

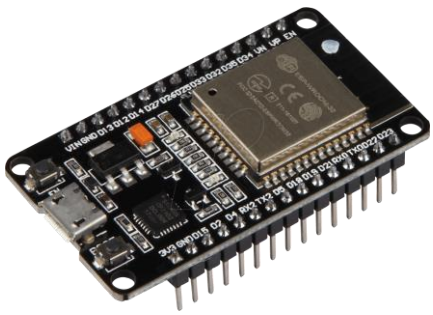


Gambar 2.2 NodeMCU ESP8266
(Sumber: components101.com, 2020)

Board ini dilengkapi dengan beberapa pin I/O, sehingga cocok untuk pengembangan aplikasi monitoring dan *controlling* dalam proyek IoT. NodeMCU ESP8266 dapat diprogram menggunakan Arduino IDE berkat kompatibilitasnya dengan *compiler* Arduino. Secara fisik, NodeMCU ESP8266 memiliki *port* USB (*mini* USB), yang mempermudah proses pemrograman (Izzinnahadi, Murdiyantoro and Armin, 2021).

2.1.3. ESP32

ESP32 merupakan mikrokontroler yang dirancang oleh *Espressif Systems* sebagai generasi lanjutan dari ESP8266. Chip ini telah dibekali dengan modul WiFi internal, sehingga sangat ideal untuk digunakan dalam pengembangan berbagai aplikasi IoT (Muliadi, Imran and Rasul, 2020). Gambar 2.3 menampilkan bentuk fisik dari ESP32.



Gambar 2.3 ESP32
(Sumber: Ruecker, 2024)

2.1.4. DHT22

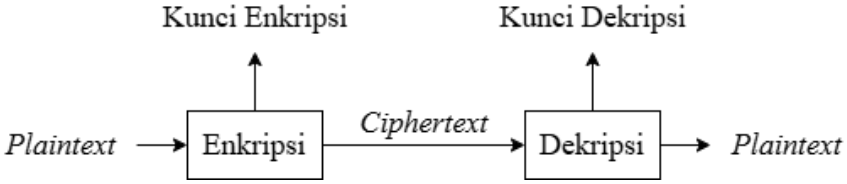
DHT22 merupakan sensor digital yang berfungsi untuk mengukur suhu dan kelembaban udara di lingkungan sekitarnya secara akurat dan presisi (Roihan *et al.*, 2021). Sensor ini menggunakan elemen kapasitif untuk mendeteksi kelembaban serta termistor untuk mengukur suhu, di mana perubahan resistansi akibat fluktuasi suhu dan kelembaban dikonversi menjadi sinyal

digital. Dengan akurasi pengukuran kelembaban sebesar ±2% dan suhu ±0.5°C, DHT22 sangat sesuai digunakan dalam berbagai aplikasi yang membutuhkan data lingkungan yang tepat dan andal. Selain tingkat akurasinya yang tinggi, sensor ini juga memiliki keunggulan dari segi kemudahan instalasi karena hanya memerlukan koneksi minimal ke mikrokontroler atau sistem pemrosesan lainnya, serta tersedia secara luas di pasaran dengan harga yang relatif terjangkau (Radya, 2024). Bentuk fisik dari sensor ini ditampilkan dalam Gambar 2.4 berikut.

Gambar 2.4 Sensor DHT22
(Sumber: Isaac, 2025)

2.1.5. Kriptografi

Kriptografi merupakan disiplin ilmu yang berfokus pada pengamanan informasi dengan cara mentransformasikan pesan asli menjadi bentuk tersandi yang tidak dapat dipahami oleh pihak-pihak yang tidak memiliki otoritas (Azhari, Mulyana, J. Perwitosari, *et al.*, 2022). Istilah "kriptografi" berasal dari bahasa Yunani, yang terdiri dari kata *"crypto"* yang berarti rahasia, dan *"graphia"* yang berarti tulisan. Dalam pengertian ini, kriptografi merupakan ilmu dan seni yang bertujuan untuk menjaga keamanan pesan saat data dikirim dari satu lokasi ke lokasi lain (Cristy and Riandari, 2021). Bidang ini mencakup pengkodean data menggunakan berbagai prosedur dan teknik matematis, yang membuatnya lebih sulit diakses atau diubah oleh pihak yang tidak berwenang (Kumar, 2024). Proses utama dalam kriptografi ditampilkan dalam Gambar 2.5 berikut.



Gambar 2.5 Proses kriptografi
(Sumber: Hidayatulloh et al., 2023)

Ada dua proses utama dalam kriptografi yaitu enkripsi dan dekripsi. Enkripsi adalah proses yang menyembunyikan data pesan dengan mengubah pesan asli menjadi bentuk yang sulit dikenali, sementara dekripsi adalah proses yang berlawanan, yang bertujuan untuk menafsirkan pesan agar dapat dibaca dengan jelas oleh pengguna (Purnamasari, Dewi and Trisetiyanto, 2021). Pesan yang telah dienkripsi disebut *ciphertext*, sedangkan pesan asli yang dapat dibaca sebelum enkripsi disebut *plaintext*. Dalam kriptografi, membedakan *plaintext* (teks asli) dan *ciphertext* (teks tersandi) adalah hal mendasar untuk menentukan tingkat kerahasiaan informasi. Proses konversi antara keduanya memanfaatkan algoritma dengan kunci rahasia yang harus sama pada tahap enkripsi dan dekripsi (Azhari et al., 2022).

Kriptografi dapat dibedakan menjadi dua kategori berdasarkan jenis kuncinya, yaitu kriptografi simetris dan kriptografi asimetris (Andriyanto and Sukmasetya, 2022). Pada kriptografi simetris, kunci yang digunakan untuk proses enkripsi dan dekripsi adalah sama. Sementara itu, kriptografi asimetris menggunakan dua kunci yang berbeda yaitu kunci *public* yang digunakan untuk proses enkripsi dan kunci *privat* yang digunakan untuk dekripsi (Saputro, Hidayati and Ujianto, 2020). Contoh algoritma yang termasuk dalam kategori simetris adalah DES (*Data Encryption Standard*), *Blowfish*, *Twofish*, MARS, IDEA, 3DES (menerapkan DES tiga kali), dan AES (*Advanced Encryption Standard*), yang dikenal sebagai *Rijndael*. Sedangkan contoh algoritma dalam kategori asimetris yaitu RSA (*Rivest Shamir Adleman*) dan ECC (*Elliptic Curve Cryptography*). Setiap jenis kriptografi memiliki kelebihan dan kelemahan yang berbeda dalam konteks meningkatkan keamanan sistem informasi (Arif and Nurokhman, 2023).

Penggunaan kriptografi sangat penting dalam berbagai bidang, seperti keamanan sistem komputer, transmisi informasi, transaksi elektronik, dan aplikasi lain yang menuntut perlindungan data bersifat rahasia (Ramalinda, Jayadi and Raharja, 2024). Selain itu, kriptografi berfungsi untuk mencegah penyangkalan, yaitu situasi di mana pengirim menolak telah mengirim pesan, sementara penerima juga tidak mengakui telah menerima pesan tersebut (Hidayatulloh *et al.*, 2023).

2.1.6. *Data Encryption Standard* (DES)

Data Encryption Standard (DES) merupakan algoritma kriptografi yang telah menjadi standar *de facto* untuk enkripsi data selama bertahun-tahun. Algoritma ini diperkenalkan oleh NIST (*National Institute of Standards and Technology*) pada tahun 1977 dan dikembangkan oleh tim IBM yang dipimpin oleh Horst Feistel dengan dukungan dari NSA (*National Security Agency*) (Simmons, 2025). DES menggunakan metode enkripsi kunci simetris, di mana kunci yang sama dipakai untuk mengenkripsi data dan untuk mengembalikan data ke bentuk semula (Permana and Nurnaningsih, 2020). Panjang kunci yang digunakan dalam algoritma ini adalah 56-bit, meskipun secara total mencapai 64-bit. Delapan bit tambahan digunakan sebagai bit paritas untuk pengecekan kesalahan, sehingga hanya kunci 56-bit yang efektif dalam proses enkripsi (Pratama *et al.*, 2023).

Menurut Rismayani dan Susanto (2020) Algoritma DES melibatkan tiga tahapan utama, baik dalam enkripsi maupun dekripsi, yaitu sebagai berikut:

1. Blok *plaintext* awal mula-mula diproses dengan matriks permutasi awal (*Initial Permutation*/IP), yang bertujuan untuk merombak urutan bit di dalam blok tersebut sebelum masuk ke tahap enkripsi.
2. Hasil dari permutasi awal tersebut kemudian dienkripsi dalam 16 putaran yang berbeda. Setiap putaran menerapkan serangkaian operasi enkripsi kompleks dan menggunakan kunci internal yang unik untuk setiap siklus.
3. Setelah melalui semua putaran enkripsi, hasil akhirnya dimutasi kembali menggunakan matriks permutasi invers (*Initial Inverse Permutation*/IP^{-1}). Proses ini mengembalikan bit ke urutan semula namun dalam bentuk *ciphertext* yang telah terenkripsi, sehingga blok *ciphertext* final siap untuk disimpan atau ditransmisikan (Pasogit, 2021).

2.1.7. *Gray Code*

Gray Code adalah sistem penomoran biner yang memungkinkan dua nilai berurutan berbeda hanya pada satu bit, mengurangi risiko kesalahan dan gangguan dalam sistem digital. Dikenal juga sebagai "kode biner terpantul" atau "kode jarak satuan", kode ini pertama kali diperkenalkan oleh Frank Gray pada tahun 1953 dan telah menjadi elemen penting dalam pendeteksian

serta perbaikan kesalahan, terutama pada komunikasi digital dan penyimpanan data. Ciri khas *Gray Code* adalah sifat reflektifnya, di mana urutan kode dapat diperluas dengan mencerminkan urutan sebelumnya sebelum menambahkan bit baru, memastikan urutan bersifat siklik dan mendukung perubahan bit tunggal. Selain *Gray Code* terpantul biner yang paling umum, terdapat varian lainnya seperti *Gray Code* seimbang (dengan transisi jumlah yang sama), *Gray Code* n-ary (untuk nilai *non-Boolean*), *Gray Code* dua dimensi (membantu koreksi kesalahan), dan *Gray Code* berurutan (sering digunakan dalam robotika dan manufaktur). Konversi *Gray Code* dari biner, yang dilakukan melalui operasi XOR pada bit paling signifikan, sangat penting dalam aplikasi yang membutuhkan akurasi tinggi, seperti konverter analog ke digital (Y, 2024). Perbedaan antara biner biasa dan *Gray Code* disajikan dalam Tabel 2.1 berikut.

Tabel 2.1 Perbedaan angka pada biner dan *Gray Code*

Desimal	*Biner*	*Gray Code*
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101

Adapun proses konversi dari *Gray Code* ke biner dapat dijelaskan sebagai berikut: Misalkan terdapat bilangan desimal 7 yang direpresentasikan dalam bentuk biner sebagai 0111. Untuk mengubahnya menjadi *Gray Code*, langkah-langkahnya adalah:

1. Pertahankan bit pertama dari bilangan biner ke *Gray Code*, (jadi untuk bit pertama dari 0111 yaitu 0).
2. Untuk bit-bit berikutnya, lakukan operasi XOR antara bit sebelumnya di biner dengan bit saat ini:
   a) Bit pertama: tetap 0.
   b) Bit kedua: XOR antara bit pertama (0) dan kedua (1) = 1.

c) Bit ketiga: XOR antara bit kedua (1) dan ketiga (1) = 0.
d) Bit keempat: XOR antara bit ketiga (1) dan keempat (1) = 0.

Jadi diperoleh *Gray Code* dari biner 0111 yaitu 0100. Adapun jika ingin melakukan konversi sebaliknya atau lebih dikenal dengan *Inverse Gray Code*, maka dapat dilakukan dengan cara berikut:

1. Pertahankan bit pertama dari bilangan *Gray Code*, (jadi untuk bit pertama biner dari 0100 yaitu 0).
2. Untuk bit-bit berikutnya, lakukan operasi penjumlahan antara bit sebelumnya di *Gray Code* dengan bit saat ini:
   a) Bit pertama: tetap 0.
   b) Bit kedua: jumlahkan antara bit pertama biner (0) dan bit kedua *Gray Code* (1) = 1.
   c) Bit ketiga: jumlahkan antara bit kedua biner (1) dan bit ketiga *Gray Code* (0) = 1.
   d) Bit keempat: jumlahkan antara bit ketiga biner (1) dan bit keempat *Gray Code* (0) = 1.

Jadi diperoleh hasil *Inverse Gray Code* 0100 menjadi 0111 dalam biner.

2.1.8. Pengujian

Pengujian merupakan salah satu tahapan krusial dalam proses pengembangan perangkat lunak, karena pengujian berperan penting dalam menjamin bahwa sistem berjalan sesuai dengan kebutuhan dan spesifikasi yang telah ditetapkan (Hendartie, Jayanti and Sutejo, 2023) Adapun menurut KBBI Daring (Badan Pengembangan dan Pembinaan Bahasa, 2025), pengujian merujuk pada proses atau cara untuk menguji suatu hal. Dalam dunia komputer, ini berarti tindakan memeriksa apakah perangkat lunak dan perangkat keras berfungsi sebagaimana mestinya. Dengan adanya proses pengujian, maka setiap kekeliruan yang mungkin terjadi pada sistem dapat dideteksi sejak dini, sehingga mencegah terjadinya kesalahan saat sistem digunakan oleh pengguna akhir (Hendartie, Jayanti and Sutejo, 2023).

Berikut merupakan beberapa pengujian yang sering digunakan dalam kriptografi dan sistem IoT.

a. *Black Box Testing*

Metode pengujian *black box* adalah teknik pengujian yang mengevaluasi hasil pelaksanaan aplikasi berdasarkan data masukan yang diberikan untuk memastikan bahwa fungsionalitas aplikasi sesuai dengan kebutuhan yang ditentukan (Febrian *et al.*, 2020). *Black box testing* atau yang lebih biasa disebut sebagai pengujian perilaku (*Behavioral Testing*), merupakan metode pengujian yang bertujuan untuk mengamati hasil *input* dan *output* dari perangkat lunak tanpa perlu mengetahui bagaimana struktur kode di dalamnya. Biasanya, pengujian ini dilakukan pada tahap akhir pengembangan untuk memastikan perangkat lunak berfungsi dengan baik sesuai yang diharapkan (Setiawan, 2021).

b. *Entropi*

Entropi merupakan metode yang digunakan untuk mengukur tingkat keacakan atau ketidakteraturan pada data hasil enkripsi. Dalam konteks ini, *entropi* digunakan untuk menilai sejauh mana *ciphertext* yang dihasilkan oleh algoritma enkripsi bersifat acak dan tidak mengandung pola yang mudah ditebak. Semakin tinggi *entropi*, semakin baik kualitas enkripsinya (Nuraeni, Agustin and Purnama, 2020; Soleh, Hadiana and Kasyidi, 2024). Penghitungan *entropi* dilakukan menggunakan rumus *Shannon Entropy*, yang disajikan dalam Persamaan (2.1) berikut.

$$H(x) = -\sum_{i=1}^{n} P(x_i) \log_2 P(x_i) \tag{2.1}$$

Keterangan:

H(x) = Nilai *entropi* dari variabel acak x (dalam satuan bit per simbol).

n = Jumlah simbol unik yang terdapat dalam data

$P(x_i)$ = Probabilitas munculnya karakter x_i dalam data.

Rumus ini menghitung rata-rata ketidakpastian dari simbol-simbol dalam suatu data. Semakin merata distribusi simbol dalam data, semakin tinggi nilai *entropi*-nya, yang menunjukkan sifat keacakan yang baik.

c. *Character Error Rate (CER)*

Character Error Rate (CER) digunakan untuk mengukur seberapa akurat proses dekripsi dalam memulihkan data asli. CER menunjukkan rasio

karakter yang salah atau rusak dibandingkan dengan total karakter yang diharapkan. Nilai CER yang rendah menandakan bahwa sistem memiliki tingkat akurasi tinggi dalam mengelola data terenkripsi (Karima *et al.*, 2024). Untuk menghitung nilai CER digunakan Persamaan (2.2) berikut.

$$CER = \frac{Jumlah\ karakter\ salah}{Total\ karakter} \text{ x } 100\% \quad (2.2)$$

2.2. Penelitian Terkait

Dalam penelitian ini, penulis merujuk pada beberapa penelitian terdahulu sebagai dasar untuk memperoleh pemahaman yang lebih mendalam, mengumpulkan informasi, serta mengidentifikasi peluang pengembangan lebih lanjut. Oleh karena itu, sejumlah referensi dari studi-studi sebelumnya yang relevan dengan topik penelitian ini telah dikumpulkan dan dianalisis. Rincian penelitian terkait disajikan dalam Tabel 2.2 berikut.

Tabel 2.2. Penelitian terkait

No.	Peneliti dan Judul	Hasil	Keterkaitan
1.	Mahesti, Ciptaningtyas and Astungkara (2025) "Penggunaan Algoritma Kriptografi DES, RSA, Modifikasi DES dan Modifikasi RSA untuk Penyandian Database"	Penelitian ini menunjukkan bahwa algoritma RSA lebih cepat dalam proses enkripsi dibanding DES, namun DES unggul dalam hal keamanan karena perhitungannya yang lebih kompleks. Modifikasi DES menghasilkan nilai ASCII rendah yang tidak terlihat (*non-visible*), sehingga kurang efektif. Sebaliknya, modifikasi RSA dengan pemilihan bilangan prima besar berhasil mempertahankan karakter ASCII yang terbaca (Mahesti,	Menggunakan algoritma DES, namun DES yang digunakan merupakan hasil modifikasi.

No.	Peneliti dan Judul	Hasil	Keterkaitan
		Ciptaningtyas and Astungkara, 2025).	
2.	Akbar, Bahri and Nirmala (2024) "Implementasi Algoritma RSA untuk Proses Enkripsi-Autentikasi Publish-Subscribe pada Protokol MQTT Menggunakan ESP8266 Berbasis IoT"	Pengujian algoritma RSA dalam MQTT menunjukkan keberhasilan menjaga keaslian pesan, dengan *error* 0,006% untuk data suhu dan 0,272% untuk data kelembaban. Keaslian pesan mencapai 99,728% hingga 99,994%. Gangguan jaringan internet menyebabkan data dari ESP8266 tidak terkirim sempurna, memicu penumpukan pesan dan perbedaan nilai (Akbar, Bahri and Nirmala, 2024).	Menggunakan protokol, mikrokontroler, dan sensor yang sama, tapi algoritma yang berbeda, yaitu RSA.
3.	Al-hazaimeh et al. (2023) "Analytical Approach for Data Encryption Standard Algorithm"	Penelitian ini menghasilkan data komparasi antara DES dan 3DES, hasilnya menunjukkan bahwa kecepatan DES lebih tinggi dibandingkan dengan 3DES, namun keamanannya lebih rendah (Al-hazaimeh et al., 2023).	Menggunakan algoritma yang sama yakni DES. Namun penelitian ini hanya berfokus pada komparasi antara DES original dan 3DES.
4.	Sari, Rosyida Zain and Cakraningrat (2022) "Implementasi Enkripsi Dan	Pengujian terhadap sistem komunikasi mencatat rata-rata *delay* sebesar 25 detik, dihitung dari waktu penerimaan data oleh *Antares* hingga berhasil	Menggunakan protokol yang sama tapi jenis board yang berbeda, yakni menggunakan

No.	Peneliti dan Judul	Hasil	Keterkaitan
	Dekripsi Pengiriman Paket Data Pada Rancang Bangun Smart Home Menggunakan Protokol MQTT"	dijalankan oleh Sistem *Smart Home* (Sari, Rosyida Zain and Cakraningrat, 2022).	*Raspberry Pi 3b+* dan algoritma yang berbeda yakni *Caeshar Cipher.*
5.	Laia, Zamzami and Sutarman (2021) "Analysis of Combination Algorithm Data Encryption Standard (DES) and Blum-Blum-Shub (BBS)"	Kombinasi DES dan *Blum-Blum Shub* (BBS) menghasilkan kunci unik yang meningkatkan keamanan tanpa mengorbankan efisiensi. Prosesnya cepat, mudah digunakan, dan cocok untuk aplikasi dengan kebutuhan keamanan tinggi (Laia, Zamzami and Sutarman, 2021).	Menggunakan algoritma DES dengan kombinasi BBS.
6.	Seif and Alexan (2020) "A High Capacity Gray Code Based Security Scheme for Non-Redundant Data Embedding"	Penggabungan algoritma *Blowfish* untuk kriptografi dengan steganografi berbasis *Gray Code* menunjukkan peningkatan kapasitas penyisipan, kualitas gambar tinggi, dan efisiensi ekstraksi lebih baik dibandingkan metode lain, membuktikan kombinasi ini efektif meningkatkan keamanan dan kapasitas data sensitif (Seif and Alexan, 2020).	Menggunakan teknik pengkodean yang sama yakni *Gray Code* tapi pada algoritma yang berbeda, yakni algoritma *blowfish*.

No.	Peneliti dan Judul	Hasil	Keterkaitan
7.	Buulolo and Sindar (2020) "Analisis dan Perancangan Keamanan Data Teks Menggunakan Algoritma Kriptografi DES (Data Encryption Standard)"	Penelitian ini menyimpulkan bahwa proses penyandian pesan dengan algoritma DES dimulai dengan mengubah *plaintext* dan *key* ke format biner. Hasil setiap putaran digabungkan kembali, lalu dimasukkan ke tabel *Permutation Compression* 2 (PC-2), yang mengompresi data CiDi dari 56-bit menjadi 48-bit (Buulolo and Sindar, 2020).	Menggunakan algoritma yang sama yakni DES. Namun penelitian ini hanya berfokus pada algoritmanya.
8.	Irawan and Winarno (2020) " Kombinasi Algoritma Kriptografi AES dan DES untuk Enkripsi File Dokumen Proposal"	Hasil implementasi menunjukkan bahwa file acak berukuran 1 MB, 204 KB, dan 159 KB mengalami peningkatan ukuran sekitar 0,05% setelah proses enkripsi dan dekripsi, disebabkan oleh penambahan bit selama enkripsi. Pengujian *Avalanche Effect* pada kombinasi algoritma AES dan DES mengindikasikan tingkat keamanan sebesar 46,38% (Irawan and Winarno, 2020).	Menggunakan algoritma DES yang dikombinasikan dengan AES namun pengimplementasianya bukan pada protokol MQTT.
9.	Ruziq, Sihombing and Sawaluddin (2020)	Kombinasi algoritma DES dan *Lattice-Based Universal Cryptography* (LUC)	Menggunakan algoritma DES

No.	Peneliti dan Judul	Hasil	Keterkaitan
	"Combination Analysis of Data Encryption Standard (DES) Algorithm and LUC Algorithm on File Security"	meningkatkan keamanan dengan menghasilkan cipher key yang lebih panjang dan sulit ditebak, sehingga lebih tahan terhadap serangan kriptografi (Ruziq, Sihombing and Sawaluddin, 2020).	dengan kombinasi LUC.
10.	Fachri and Sembiring (2020) "Pengamanan Data Teks Menggunakan Algoritma DES Berbasis Android"	Hasil penelitian menunjukkan bahwa aplikasi yang dirancang menggunakan algoritma DES berhasil mengamankan pesan teks melalui enkripsi, sehingga hanya pihak yang memiliki kunci yang dapat membaca pesan tersebut. Proses dekripsi juga berfungsi dengan baik, memungkinkan pesan terenkripsi dikembalikan ke bentuk asli. Secara keseluruhan, aplikasi ini efektif dalam menjaga kerahasiaan pesan dan mengatasi potensi ancaman keamanan.	Menggunakan algoritma DES namun berbasis Android.

BAB V
PENUTUP

4.1. KESIMPULAN

Penelitian ini berhasil mengimplementasikan algoritma *Data Encryption Standard* (DES) dan teknik *Gray Code* pada protokol *Message Queuing Telemetry Transport* (MQTT) untuk memperkuat keamanan komunikasi data pada sistem *Internet of Things* (IoT). Sistem yang dibangun mampu mengintegrasikan sensor DHT22 dengan mikrokontroler ESP8266 dan ESP32 serta menggunakan MQTT sebagai media pengiriman data terenkripsi. Pada sisi pengembangan, penggunaan dua *library* khusus, yaitu *"GrayCodeToDES"* untuk konversi *Gray Code* dan enkripsi DES, serta *"DESGrayDecoder"* untuk dekripsi dan *Inverse Gray Code*, berhasil membuat sistem menjadi lebih modular, terstruktur, dan stabil tanpa bergantung pada *library* eksternal.

Berdasarkan hasil pengujian, sistem menunjukkan performa fungsionalitas yang sangat baik. Dari sisi akurasi, pengujian *black box* membuktikan bahwa data berhasil dikirim dan diterima dengan akurasi 100%, termasuk kemampuan mendeteksi *error* saat sensor dilepas dan menangani koneksi *broker* MQTT yang terputus. Pada pengujian *delay* pengiriman data MQTT, ESP8266 pada interval tanpa jeda (0 ms) memiliki rata-rata *delay* sebesar 1826 ms. Saat interval pengiriman diatur menjadi 1000 ms, rata-rata *delay* turun drastis menjadi 160 ms. Adapun saat interval 60000 ms, rata-rata *delay* meningkat menjadi 251 ms. Sementara itu, ESP32 menunjukkan performa yang lebih konsisten, dengan rata-rata *delay* sebesar 247 ms pada interval 0 ms, 220 ms pada 1000 ms, dan 213 ms pada 60000 ms. Hasil ini membuktikan bahwa ESP32 lebih stabil dan responsif dalam pengiriman data dibandingkan ESP8266.

Pada pengujian pemrosesan *Gray Code*, ESP8266 menunjukkan rata-rata waktu proses sebesar 1,555 ms (0 ms), 1,625 ms (1000 ms), dan 1,732 ms (60000 ms). Sementara ESP32 memiliki rata-rata waktu proses 0,714 ms (0 ms), 0,750 ms (1000 ms), dan 0,763 ms (60000 ms), menunjukkan kinerja yang jauh lebih cepat dan stabil. Pada proses enkripsi DES, ESP8266 mencatat rata-rata waktu 2,498 ms (0 ms), 2,598 ms (1000 ms), dan 2,618 ms (60000 ms), sedangkan ESP32 secara

konsisten lebih unggul dengan rata-rata 0,718 ms (0 ms), 0,725 ms (1000 ms), dan 0,740 ms (60000 ms). Untuk dekripsi di sisi *subscriber*, data dari ESP8266 memiliki rata-rata waktu 0,888 ms (0 ms), 0,959 ms (1000 ms), dan 1,019 ms (60000 ms). Sedangkan untuk data dari ESP32, waktu dekripsi tercatat sebesar 0,967 ms (0 ms), 0,900 ms (1000 ms), dan 1,073 ms (60000 ms), menunjukkan stabilitas waktu proses yang lebih baik dibandingkan ESP8266. Dalam pengujian waktu *Inverse Gray Code* setelah dekripsi, waktu proses untuk data ESP8266 dan ESP32 berada di kisaran rata-rata 0,038–0,042 ms, baik pada interval 0 ms, 1000 ms, maupun 60000 ms, menegaskan bahwa proses *Inverse Gray Code* ini sangat ringan dan tidak menjadi *bottleneck* pada sistem.

Pada sisi keamanan, pengujian *entropi* menunjukkan bahwa *ciphertext* dari ESP8266 memiliki rata-rata *entropi* 3,157 (0 ms), 3,017 (1000 ms), dan 3,217 (60000 ms), sedangkan *ciphertext* dari ESP32 mencatat *entropi* rata-rata 3,291 (0 ms), 3,269 (1000 ms), dan 3,162 (60000 ms). Seluruh nilai *entropi* mendekati batas maksimal teoritis untuk 64-bit, yaitu 4,0. Hal ini menunjukkan tingkat keacakan *ciphertext* yang sangat baik. Lebih jauh, pengujian *Character Error Rate* (CER) menunjukkan hasil sempurna, yaitu 0% untuk semua skenario dan semua interval waktu, membuktikan bahwa transmisi data berlangsung tanpa kesalahan karakter apapun.

Terakhir, dari aspek memori, ESP8266 menunjukkan *heap* awal sekitar 49352 *byte* dengan penggunaan RAM rata-rata 1958 *byte* (0 ms), 639 *byte* (1000 ms), dan 1196 *byte* (60000 ms). Sedangkan ESP32 memiliki *heap* awal dengan variasi berbeda yaitu 237288 *byte* (0 ms), 237284 *byte* (1000 ms), dan 237448 *byte* (60000 ms) dengan penggunaan RAM rata-rata 868 *byte* (0 ms), 810 *byte* (1000 ms), dan 974 *byte* (60000 ms), menunjukkan bahwa ESP32 lebih efisien dan stabil dalam pengelolaan memori.

Secara keseluruhan, implementasi algoritma DES dan *Gray Code* dalam protokol MQTT telah terbukti meningkatkan keamanan, kecepatan, dan keandalan sistem komunikasi IoT. Dengan hasil pengujian yang menunjukkan performa lebih stabil dan efisien, ESP32 direkomendasikan untuk aplikasi IoT berskala besar atau yang memerlukan kecepatan tinggi dan kestabilan memori jangka panjang,

sedangkan ESP8266 tetap dapat digunakan untuk skenario yang lebih sederhana atau dengan kebutuhan sumber daya yang lebih ringan.

4.2. SARAN

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan selama penelitian, penulis memiliki beberapa saran untuk penelitian selanjutnya yaitu sebagai berikut.

1. Melakukan pengujian dengan data yang lebih banyak dan beragam jenis interval, agar hasil yang diperoleh lebih akurat.
2. Mengembangkan algoritma DES standar dengan berbagai macam gerbang logika, baik di tahapan awal maupun dalam *Feistel Network*.
3. Melakukan eksperimen dengan mengombinasikan algoritma DES dengan algoritma kriptografi lainnya.

DAFTAR PUSTAKA

Abduh, M. (2021) *Contactless Temperature Screening Berbasis IoT Menggunakan Metode Failover dan Protokol MQTT*. Universitas Islam Negeri Syarif Hidayatullah.

Pratama, A. *et al.* (2023) 'Algoritma DES (Data Encryption Standard) untuk Keamanan Digital', *SITEBA*, 2(1), pp. 15–18.

Ahyadi, Z. *et al.* (2021) 'Sistem IoT Untuk Monitoring Penggunaan Energi Listrik Dengan Protokol MQTT', *Jurnal Poros Teknik*, 13(1), pp. 52–58.

Akbar, R. *et al.* (2023) 'Experimental Reseacrch Dalam Metodologi Pendidikan', *Jurnal Ilmiah Wahana Pendidikan*, 9(2), pp. 465–474.

Akbar, R., Bahri, S. and Nirmala, I. (2024) 'Implementasi Algoritma RSA untuk Proses Enkripsi-Autentikasi Publish-Subscribe pada Protokol MQTT Menggunakan ESP8266 Berbasis IoT', *Coding: Jurnal Komputer dan Aplikasi*, 12(1), pp. 23–32.

Alfasa, F.P., Dewanta, F. and Istikmal (2024) 'Implementasi MQTT Sebagai Protokol Komunikasi Pada Prototipe Sistem Monitoring Smart Building', *e-Proceeding of Engineering*, 11(2), pp. 1182–1188.

Al-hazaimeh, O.M. *et al.* (2023) 'Analytical Approach for Data Encryption Standard Algorithm', *International Journal of Interactive Mobile Technologies*, 17(14), pp. 126–143.

Alshammari, H.H. (2023) 'The internet of things healthcare monitoring system based on MQTT protocol', *Alexandria Engineering Journal*, 69, pp. 275–287.

Andriyanto, M.R. and Sukmasetya, P. (2022) 'Penerapan Algoritma Advanced Encryption Standard (AES) Untuk Keamanan Data Transaksi Pada Sistem E-Marketplace', *Journal of Computer System and Informatics (JoSYC)*, 4(1), pp. 179–187.

Apinento, F.B.P.J.A., Abdan, M.K. and Riel, J. (2023) 'Analisis Komprehensif terhadap Ancaman Saat Ini dan Penanggulangan yang Efektif', in *Meningkatkan Keamanan Siber*. Pontianak, pp. 1–13.

Arif, Z. and Nurokhman, A. (2023) 'Analisis Perbandingan Algoritma Kriptografi Simetris Dan Asimetris Dalam Meningkatkan Keamanan Sistem Informasi', *JTSI*, 4(2), pp. 394–405.

Azhari, M., Mulyana, D.I., Perwitosari, J., *et al.* (2022) 'Implementasi Pengamanan Data pada Dokumen Menggunakan Algoritma Kriptografi Advanced

Encryption Standard (AES)', *Jurnal Pendidikan Sains dan Komputer*, 2(1), pp. 163–171.

Azhari, M., Mulyana, D.I., Perwitosari, F.J., *et al.* (2022) 'Implementasi Pengamanan Data pada Dokumen Menggunakan Algoritma Kriptografi Advanced Encryption Standard (AES)', *Jurnal Pendidikan Sains dan Komputer*, 2(1), pp. 163–171.

Badan Pengembangan dan Pembinaan Bahasa (2025) *Pengujian*, *Kamus Besar Bahasa Indonesia (KBBI) Daring*. Available at: https://kbbi.kemdikbud.go.id/entri/pengujian (Accessed: 13 February 2025).

Bouyuklieva, S. *et al.* (2024) 'Generating m-Ary Gray Codes and Related Algorithms', *Algorithms*, 17(7), pp. 1–20.

Buulolo, N. and Sindar, A. (2020) 'Analisis dan Perancangan Keamanan Data Teks Menggunakan Algoritma Kriptografi DES (Data Encryption Standard)', *Jurnal Teknologi Informasi*, 15(3), pp. 61–65.

Cristy, N. and Riandari, F. (2021) 'Implementasi Metode Advanced Encryption Standard (AES 128 Bit) Untuk Mengamankan Data Keuangan', *JIKOMSI [Jurnal Ilmu Komputer dan Sistem Informasi]*, 4(2), pp. 75–85.

Diono, M. *et al.* (2021) 'Sistem Monitoring Jaringan Sensor Node Berbasis Protokol MQTT', *Jurnal Politeknik Caltex Riau*, 7(2), pp. 120–126.

Fachri, B. and Sembiring, R.M. (2020) 'Pengamanan Data Teks Menggunakan Algoritma DES Berbasis Android', *Jurnal media Informatika Budidarma*, 4(1), pp. 110–116.

Febrian, V. *et al.* (2020) 'Pengujian pada Aplikasi Penggajian Pegawai dengan menggunakan Metode Blackbox', *Jurnal Informatika Universitas Pamulang*, 5(1), pp. 61–66.

Fikhri, A.A. and Nurdin, N. (2024) 'Implementasi Algoritma K-Nearest Neighbor Pada Sistem Pemantau Suhu dan Kelembapan Ruang Server Menggunakan Protokol MQTT Berbasis IoT', *Jurnal Informatika dan Teknik Elektro Terapan*, 12(3S1), pp. 4586–4596.

Handy, M.V. and Pardede, M. (2022) 'Implementasi Protokol MQTT dan Nodered Untuk Pemantauan Suhu dan Kelembapan Ruang Kelas Berbasis IoT', *TRekRiTel*, 2(2), pp. 1–15.

Hanif, A.A. and Ilyas, M. (2024) 'Effective Feature Engineering Framework for Securing MQTT Protocol in IoT Environments', *Sensors*, 24(6), pp. 1–21.

Harnanta, K.J., Bhawiyuga, A. and Basuki, A. (2020) 'Implementasi MQTT Broker dengan Kemampuan Auto Scaling pada Internet of Things', *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 4(6), pp. 1783–1792.

Hendartie, S., Jayanti, S. and Sutejo, H. (2023) ‘Pengujian Aplikasi Penerimaan Mahasiswa Baru (PMB) STMIK Palangkaraya Menggunakan Black Box Testing (Testing the STMIK Palangkaraya New Student Admission Application Using Black Box Testing)’, *Jurnal Sains Komputer dan Teknologi Informasi*, 5(2), pp. 31–40.

Hidayatulloh, N.W. *et al.* (2023) ‘Mengenal Advance Encrytion Standard (AES) Sebagai Algoritma Kriptografi Dalam Mengamankan Data’, *Digital Transformation Technology (Digitech) | e*, 3(1), pp. 1–10.

Im, Y. and Lim, M. (2023) ‘E-MQTT: End-to-End Synchronous and Asynchronous Communication Mechanisms in MQTT Protocol’, *Applied Science*, 13(22).

Irawan, C. and Winarno, A. (2020) ‘Kombinasi Algoritma Kriptografi AES dan DES untuk Enkripsi File Dokumen Proposal’, in *Proceeding SENDIU*. Semarang, pp. 28–35.

Isaac (2025) *DHT22 - sensor suhu dan kelembaban presisi*, *Hardwarelibre*. Available at: https://id.hwlibre.com/dht22/ (Accessed: 12 April 2025).

Izzinnahadi, A., Murdiyantoro, R.A. and Armin, E.U. (2021) ‘Sistem Pemantauan Kondisi Air Hidroponik Berbasis Internet of Things Menggunakan NodeMCU ESP8266’, *Journal of Telecommunication, Electronics, and Control Engineering (JTECE)*, 3(2), pp. 54–61.

Karima, N.A. *et al.* (2024) ‘Kriptografi Teks Berbasis Algoritma Substitusi Vigenere Cipher 8 Bit’, *Jurnal Masyarakat Informatika*, 15(1), pp. 1–13.

Kumar, S. (2024) *Cryptography to Quantum Cryptography Chapter*. 15th edn, *Dimension of Innovation and Technology in Rising India*. 15th edn. New Delhi: Akinik Publishers.

Laia, O., Zamzami, E.M. and Sutarman (2021) ‘Analysis of Combination Algorithm Data Encryption Standard (DES) and Blum-Blum-Shub (BBS)’, in *Journal of Physics: Conference Series*. IOP Publishing Ltd.

Mahesti, T., Ciptaningtyas, A.F. and Astungkara, A. (2025) ‘Perbandingan Penggunaan Algoritma Kriptografi DES, RSA, Modifikasi DES dan Modifikasi RSA untuk Penyandian Database’, *Jurnal Ilmiah ILKOMINFO*, 8(1), pp. 1–15.

Manullang, A.B.P., Saragih, Y. and Hidayat, R. (2021) ‘Implementasi NodeMCU ESP8266 dalam Rancang Bangun Sistem Keamanan Sepeda Motor Berbasis IoT’, *Jurnal Informatika & Rekayasa Elektronika*, 4(2), pp. 163–170. Available at: http://e-journal.stmiklombok.ac.id/index.php/jireISSN.2620-6900.

Mishra, B. and Kertesz, A. (2020) ‘The use of MQTT in M2M and IoT systems: A survey’, *IEEE Access*, 8, pp. 201071–201086.

Muliadi, Imran, A. and Rasul, Muh. (2020) 'Pengembangan Tempat Sampah Pintar Menggunakan ESP32', *Jurnal Media Elektrik*, 17(2), pp. 73–79.

Munte, S.R. *et al.* (2023) 'Jenis Penelitian Eksperimen dan Noneksperimen (Design Klausal Komparatif dan Design Korelasional)', *Jurnal Pendidikan Tambusai*, 7(3).

Nuraeni, F., Agustin, Y.H. and Purnama, A.E. (2020) 'Implementasi Caesar Cipher and Advanced Encryption Standard (AES) Pada Pengamanan Data Pajak Bumi Bangunan', *Jurnal Ilmiah MATRIK*, 22(2), pp. 187–194.

Pasogit, B. (2021) *Data Encryption Standard (DES) : Pengertian Lengkap dan Sejarah*, *Student Terpelajar*. Available at: https://www.studentterpelajar.com/2021/03/pengertian-algoritma-des.html (Accessed: 25 April 2025).

Permana, A.A. and Nurnaningsih, D. (2020) 'Application of Cryptography with Data Encryption Standard (Des) Algorithm in Picture', *JIKA (Jurnal Informatika) Universitas Muhammadiyah Tangerang*, 4(2), pp. 9–14.

Pratama, R.F., Wicaksono, R.S.R. and Pramudhita, A.N. (2023) 'Perancangan dan Implementasi Protokol MQTT pada Sistem Parkir Cerdas Berbasis IoT', *Jurnal Informatika dan Teknik Elektro Terapan*, 11(3), pp. 475–483.

Purnamasari, D., Dewi, A.K. and Trisetiyanto, A.N. (2021) 'Analisis Performansi Kriptografi Berbasis Caesar Cipher Untuk Keamanan Data Menggunakan Python Pada Tembang Macapat', *Journal of Systems, Information Technology, and Electronics Engineering*, 1(2), pp. 50–54.

Qotrun (2021) *Klasifikasi Jenis-Jenis Metode Penelitian Yang Sering Dipakai*, *Gramedia.com*.

Radya, M. (2024) *Mengukur Tingkat Kelembaban Udara: Sensor DHT22*, *Indobot Academy*. Available at: https://blog.indobot.co.id/mengukur-tingkat-kelembaban-udara-sensor-dht22/ (Accessed: 12 April 2025).

Ramalinda, D., Jayadi and Raharja, A.R. (2024) 'Strategi Perlindungan Data Menggunakan Sistem Kriptografi Dalam Keamanan Informasi', *Journal of International Multidisciplinary Research*, 4(6), pp. 665–671.

Rismayani and Susanto, C. (2020) 'Using AES and des Cryptography for System Development File Submission Security Mobile-Based', in *2020 8th International Conference on Cyber and IT Service Management, CITSM 2020*. Kuala Lumpur: Institute of Electrical and Electronics Engineers Inc., pp. 1–7.

Roihan, A. *et al.* (2021) 'Simulasi Pendeteksi Kelembaban Pada Tanah Menggunakan Sensor DHT22 Dengan Proteus', *Jurnal METHODIKA*, 7(1).

Ruecker, E. (2024) *Esp32-devkitc V4 Arduino Ide*, *finaterisrmuser.z14.web.core.windows.net*. Available at: https://finaterisrmuser.z14.web.core.windows.net/esp32-devkitc-v4-arduino-ide.html (Accessed: 10 April 2025).

Ruziq, F., Sihombing, P. and Sawaluddin (2020) 'Combination Analysis of Data Encryption Standard (DES) Algorithm and LUC Algorithm on File Security', *International Journal of Research and Review (ijrrjournal.com)*, 7(2), pp. 140–144.

Saputro, T.H., Hidayati, N. and Ujianto, E.I.H. (2020) 'Survei tentang Algoritma Kriptografi Asimetris', *Jurnal Informatika Polinema*, 6(2), pp. 67–72.

Sari, R., Rosyida Zain, A. and Cakraningrat, M.S. (2022) 'Implementasi Enkripsi Dan Dekripsi Pengiriman Paket Data Pada Rancang Bangun Smart Home Menggunakan Protokol MQTT', *Jurnal Multinetics*, 8(2), pp. 168–176.

Seif, A. and Alexan, W. (2020) 'A High Capacity Gray Code Based Security Scheme for Non-Redundant Data Embedding', in. Aswan: Institute of Electrical and Electronics Engineers Inc., pp. 130–136.

Setiawan, R. (2021) *Black Box Testing Untuk Menguji Perangkat Lunak*, *Dicoding*. Available at: https://www.dicoding.com/blog/black-box-testing/ (Accessed: 20 March 2025).

Sidik, R., Suarna, N. and Rinaldi Dikananda, A. (2023) 'Analisa Data Set Peminatan Siswa Menggunakan Algoritma K-Means dengan Optimasi Parameter di Sekolah Menengah Kejuruan (Studi Kasus: SMK PUI Gegesik)', *Jurnal Mahasiswa Teknik Informatika*, 7(2), pp. 1197–1203.

Simmons, G.J. (2025) *Data Encryption Standard*, *Britannica*. Available at: https://www.britannica.com/topic/AES (Accessed: 10 January 2025).

Soleh, M.N.Z., Hadiana, A.I. and Kasyidi, F. (2024) 'Kriptografi Homomorfik dalam Anonimisasi Data untuk Pengolahan Data pada Sistem E-Voting', *Jurnal Masyarakat Informatika*, 15(1), pp. 107–124.

Suprianto, G., Natasya, A.R. and Riskiawan, A.I. (2023) 'Sistem Pendeteksi Kebocoran Gas Berbasis IoT Sebagai Alat Bnatu Pada UMKM', *Zetroem*, 5(1), pp. 62–67.

Thahara, A. and Siregar, I.T. (2021) 'Implementasi Kriptografi untuk Keamanan Data dan Jaringan menggunakan Algoritma DES', *JURTI*, 5(1).

Y, R. (2024) *Gray Code*, *electronicsdesk.com*. Available at: https://electronicsdesk.com/gray-code.html (Accessed: 27 October 2024).