**SISTEM IOT UNTUK PENGIRIMAN DATA GAMBAR**

**Oleh**

**VIERI FAJAR FIRDAUS**

**NIM : 13521099**

**JURUSAN : TEKNIK INFORMATIKA**

A black and white drawing of an elephant

AI-generated content may be incorrect.

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO & INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**April 2025**

Pendahuluan

### Latar Belakang

Perkembangan Internet of Things (IoT) yang pesat dalam beberapa tahun terakhir telah membawa perubahan signifikan dalam berbagai sektor, mulai dari rumah pintar (smart homes), pertanian pintar (smart farming), hingga pemantauan kesehatan jarak jauh (telemedicine). IoT memungkinkan perangkat yang terhubung untuk saling bertukar data dan berinteraksi secara otomatis melalui jaringan internet. Salah satu aplikasi penting IoT adalah dalam pengolahan data berbasis gambar, yang dapat digunakan dalam berbagai bidang seperti pengawasan keamanan, analisis citra medis, dan pemantauan lingkungan.

Salah satu tantangan utama dalam aplikasi IoT yang berbasis gambar adalah bagaimana menangani pengiriman data gambar secara efisien dan aman. Data gambar berukuran besar memerlukan bandwidth yang cukup besar, dan pengiriman data secara terus-menerus, seperti dalam streaming video, bisa sangat membebani jaringan jika tidak dikelola dengan baik. Oleh karena itu, pengiriman gambar melalui protokol komunikasi yang efisien, seperti MQTT (Message Queuing Telemetry Transport), sangat penting untuk memastikan bahwa data dapat dikirimkan dengan latensi yang rendah dan menggunakan bandwidth yang efisien.

### Tujuan

* Merancang sistem pengiriman data gambar dari ESP board ke server menggunakan MQTT.
* Menyimpan data di database dan menampilkannya pada dashboard.
* Mengevaluasi kinerja sistem berdasarkan interval pengiriman dan latensi end-to-end.
* (Opsional) Mengimplementasikan enkripsi-dekripsi data untuk keamanan.

Rancangan Sistem

Spesifikasi Sistem

|  |  |
| --- | --- |
| **Komponen** | **Deskripsi** |
| **Sensing Unit** | ESP32-CAM |
| **Resolusi Gambar** | 640×480 piksel (Base64) |
| **Protokol Komunikasi** | MQTT (Wi-Fi) |
| **Interval Pengiriman (T)** | T = 10 (dalam detik) NIM=13521099 |
| **Jumlah Pengiriman (K)** | K ∈ {10, 20, 100} |
| **Data yang Dikirim** | {image\_data, timestamp, capture\_time, publish\_time} |

### Kebutuhan Sistem

Sistem IoT yang dibuat memiliki beberapa fitur dibawah ini :

* Mengirimkan gambar secara real time melalui MQTT
* Melakukan enkripsi terhadap gambar
* Melihat gambar melalui dashboard

Selain itu, Penulis menggunakan ESP32 CAM yang dapat mengirimkan gambar berdasarkan hasil yang ditangkap.

### Diagram Blok Sistem

Arsitektur sistem ini dibangun dengan pendekatan modular, yang terdiri dari beberapa komponen utama yang saling berinteraksi untuk mencapai tujuan pengiriman gambar secara efisien dan aman. Komponen utama dalam arsitektur ini adalah perangkat **ESP32** sebagai unit pengambil data, **MQTT** sebagai protokol komunikasi, **backend server** untuk pengelolaan data, dan **frontend** sebagai antarmuka pengguna untuk menampilkan gambar. Untuk mengelola service yang ada digunakan containerization menggunakan docker.

A screenshot of a computer

AI-generated content may be incorrect.

### Alur Pengiriman Data

1. ESP32 mengambil gambar dan menyimpannya dalam buffer.
2. Data gambar dikonversi ke format Base64 (jika diperlukan).
3. Data dikirim via MQTT dengan format JSON
4. Server MQTT menyimpan data ke database (MySQL).
5. Dashboard menampilkan data gambar dan informasi pengiriman.

### Implementasi Kode di ESP32

|  |
| --- |
| void sendPhoto() {    camera\_fb\_t \* fb = NULL;    start\_time = millis();    fb = esp\_camera\_fb\_get();    if(!fb) {      Serial.println("Camera capture failed");      delay(1000);      ESP.restart();    }    end\_time = millis();    capture\_time = end\_time - start\_time;    Serial.println("Connecting to MQTT server...");    if (client.connect("ESP32Client", mqtt\_user, mqtt\_pass)) {      Serial.println("Connected to MQTT server");      String base64Image = base64::encode(fb->buf, fb->len);      String payloadImage = "{\"id\":" + String(message\_id) +                           ",\"image\":\"" + base64Image + "\"}";      start\_time = millis();      client.publish(mqtt\_topic\_image, payloadImage.c\_str());      end\_time = millis();      publish\_time = end\_time - start\_time;        String payloadLatency = "{\"id\":" + String(message\_id) +                            ",\"capture\_time\":" + String(capture\_time) +                            ",\"publish\_time\":" + String(publish\_time) +"}";      client.publish(mqtt\_topic\_latency, payloadLatency.c\_str());      Serial.println("Image sent to MQTT topic.");      esp\_camera\_fb\_return(fb);      success++;    } else {      failed++;      Serial.println("Failed to connect to MQTT broker.");    }    int total = success+failed;    if(total > 0) {      float successRate = (success / float(total)) \* 100;      Serial.print("Success Rate: ");      Serial.print(successRate);      Serial.print("%");      Serial.print("(");      Serial.print(success);      Serial.print("/");      Serial.print(total);      Serial.println(")");    }    client.loop();  } |

### Algoritma Enkripsi

Enkripsi menggunakan algoritma AES, dimana proses enkripsi dilakukan ketika server menerima request berupa base64 lalu hasil enkripsi akan disimpan di database, untuk hasil dekripsi bisa diperoleh ketika user membuka dashboard

|  |
| --- |
| def encrypt\_image(image\_data):      cipher = AES.new(AES\_KEY, AES.MODE\_CBC)      ct\_bytes = cipher.encrypt(pad(image\_data, AES.block\_size))      iv = cipher.iv      return iv + ct\_bytes |

|  |
| --- |
| def decrypt\_image(encrypted\_data):      """Dekripsi gambar menggunakan AES CBC"""      iv = encrypted\_data[:16]      ct = encrypted\_data[16:]      cipher = AES.new(AES\_KEY, AES.MODE\_CBC, iv=iv)      pt = unpad(cipher.decrypt(ct), AES.block\_size)      return pt |

### Evaluasi Kinerja

Untuk latensi yang dihitung pada evaluasi kinerja dari pengambilan gambar, pengiriman melalui MQTT, proses enkripsi, proses input, untuk proses penampilan gambar pada dashboard akan diambil beberapa sampel

|  |  |  |
| --- | --- | --- |
| **K** | **Rata-rata Latency (ms)** | **Error Interval (%)** |
| 10 | 750.80 | 0 |
| 20 | 1628.02 | 0 |
| 100 | 841.65 | 0 |