SISTEM IOT UNTUK PENGIRIMAN DATA GAMBAR

Oleh

VIERI FAJAR FIRDAUS

NIM : 13521099

JURUSAN : TEKNIK INFORMATIKA

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

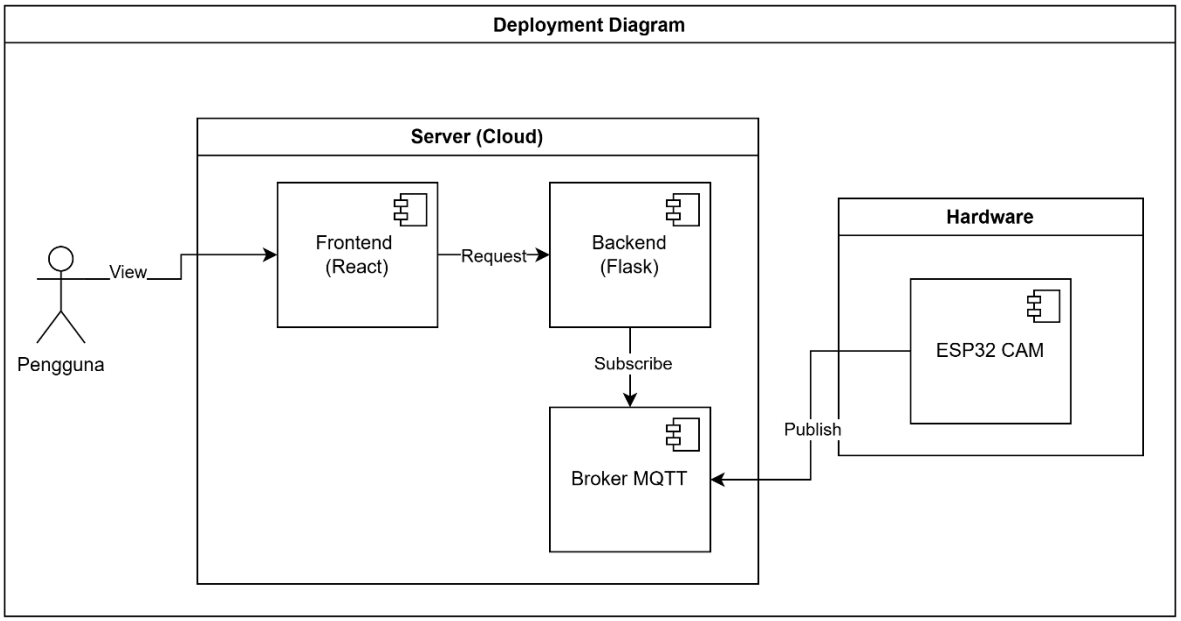
Sistem IoT yang dibuat memiliki beberapa fitur dibawah ini :

- 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.



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);

    Serial.println(getFormattedTime());
    String payloadImage = "{\"id\":" + String(message_id) +
                      ",\"image\":\"" + base64Image + "\"" +
                      ",\"timestamp\":\"" + getFormattedTime() + "\"}";
    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
```

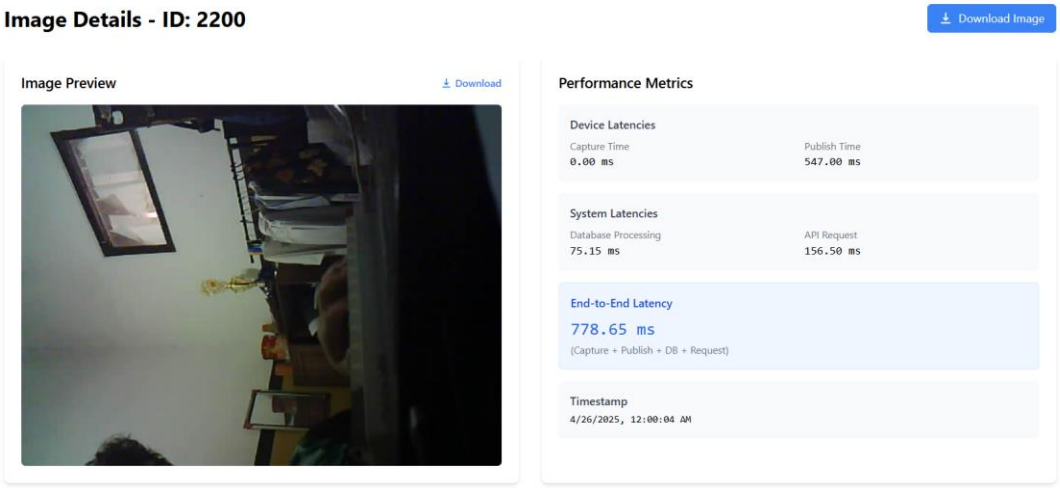
```
def decrypt_image(encrypted_data):
    """Dekripsi gambar menggunakan AES CBC"""
    iv = encrypted_data[:16]
    ct = encrypted_data[16:]
    cipher = AES.new(AES_KEY, AES.MODE_CBC, iv=iv)
    pt = unpad(cipher.decrypt(ct), AES.block_size)
    return pt
```

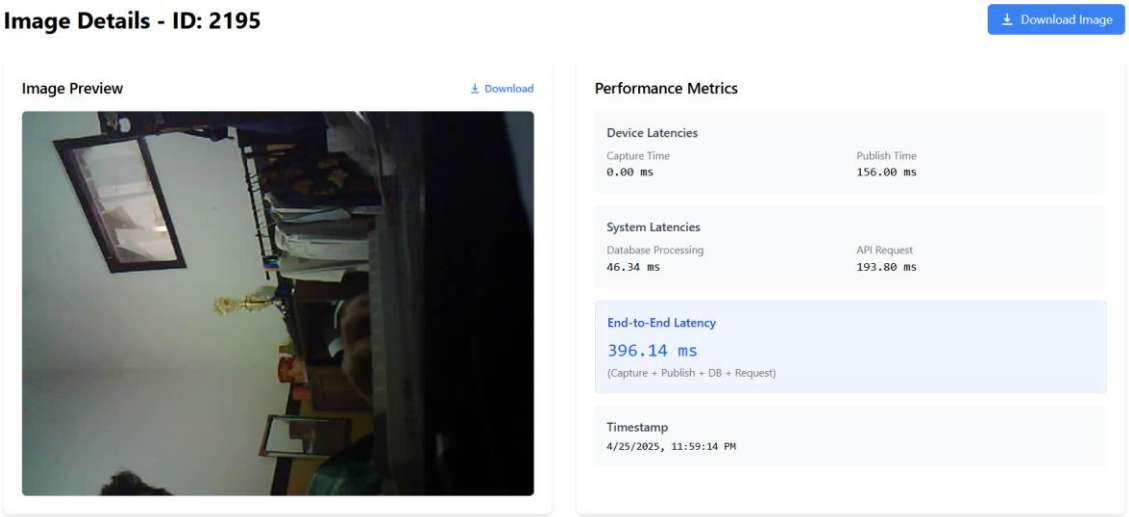
Evaluasi Kinerja

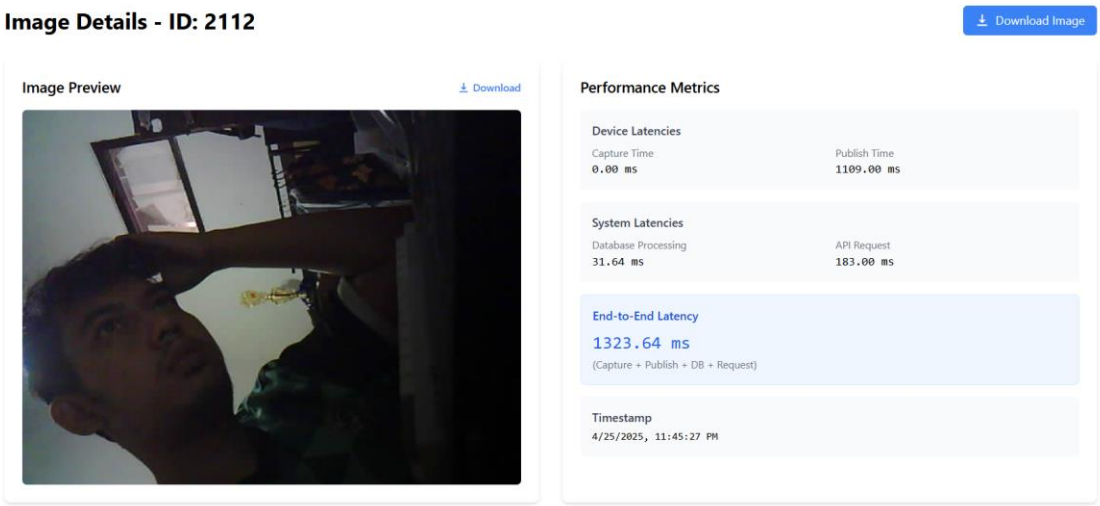
Untuk latensi yang dihitung pada evaluasi kinerja dari pengambilan gambar, pengiriman melalui MQTT, proses enkripsi, proses input, untuk proses penampilan gambar pada dashboard akan diambil beberapa sampel

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

Berikut adalah sampel dari latency end to end







Kesimpulan

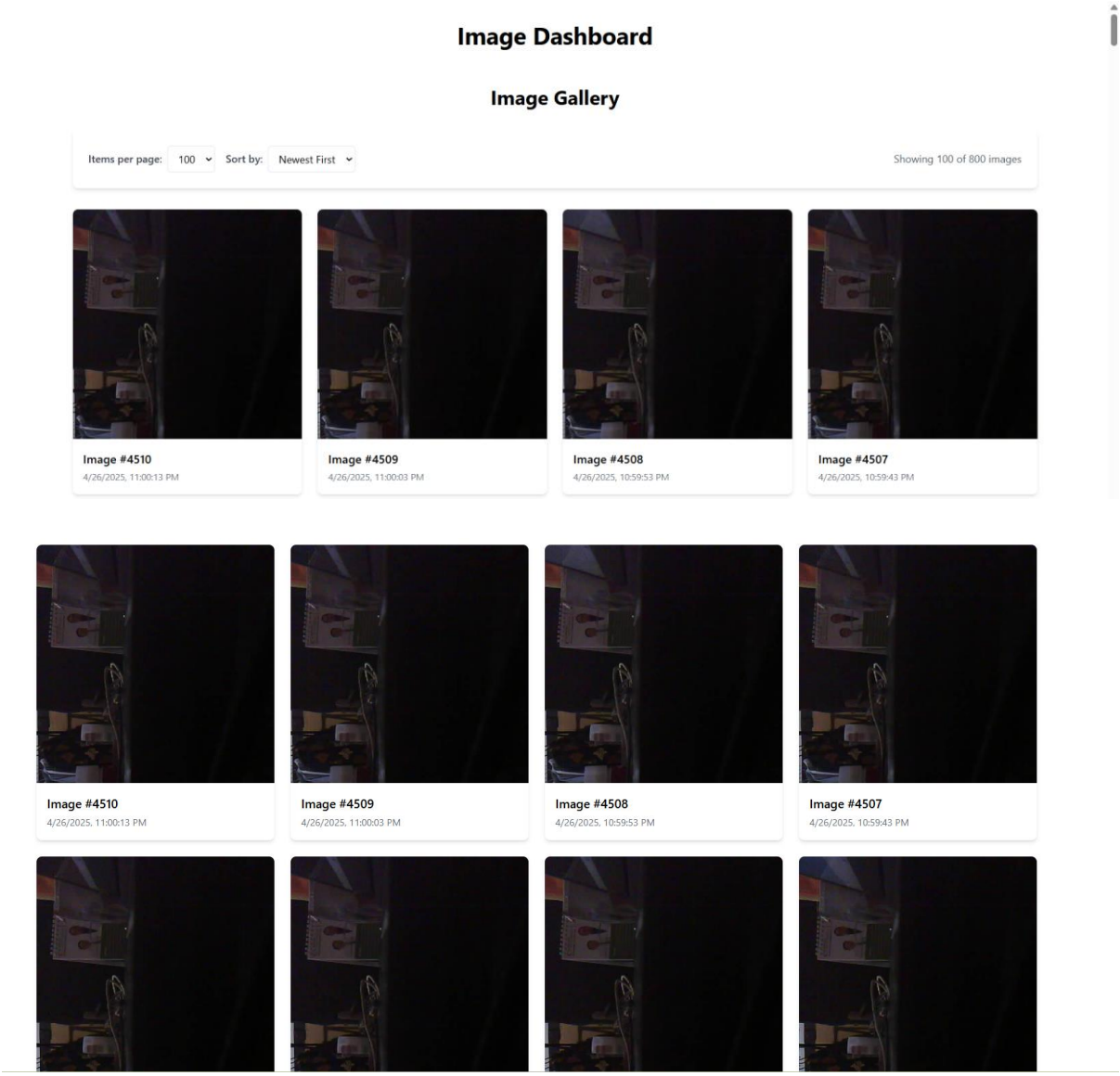
Sistem IoT yang dirancang berhasil mengirim data gambar dari ESP board ke server menggunakan protokol MQTT dengan interval **10 detik** (sesuai 3 digit terakhir NIM). Hasil pengujian menunjukkan bahwa:

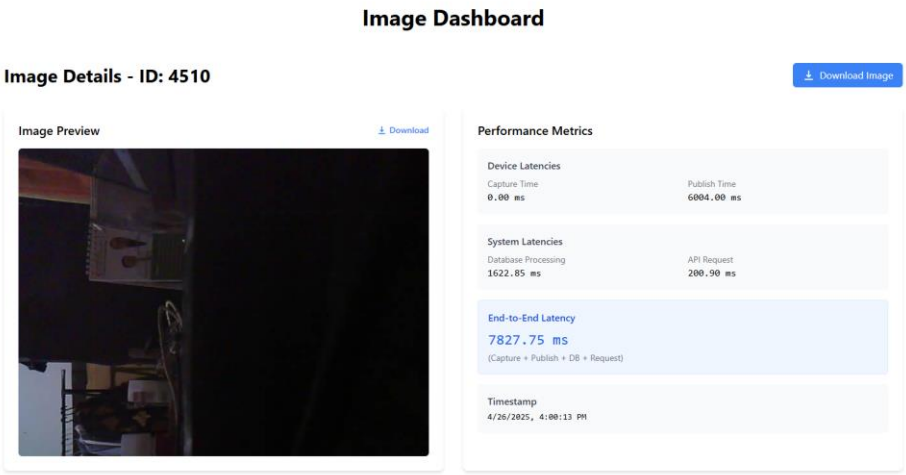
- **Latensi pengiriman** bervariasi tergantung pada jumlah pengiriman (K), dengan rata-rata terendah **750.80 ms** (saat K=10) dan tertinggi **1628.02 ms** (saat K=20). Nilai latensi kembali stabil di **841.65 ms** untuk K=100, menunjukkan bahwa sistem mampu menangani beban pengiriman berulang.
- **Error interval pengiriman** adalah **0%** untuk semua nilai `K**, membuktikan bahwa pengiriman data tepat waktu sesuai konfigurasi.
- **Enkripsi data** diimplementasikan di sisi server (menggunakan Docker untuk containerisasi), memastikan keamanan data tanpa menambah beban komputasi di perangkat ESP.
- **Docker** pada server berperan dalam mempermudah deployment dan isolasi proses, termasuk penerimaan data MQTT, enkripsi, dan penyimpanan ke database.

Secara keseluruhan, sistem memenuhi kriteria Level 1 (pengiriman data real-time) dan Level 2 (enkripsi opsional), dengan kinerja yang stabil dan skalabel. Untuk pengembangan selanjutnya, uji coba dengan **interval dinamis** atau **algoritma enkripsi di sisi ESP** dapat dipertimbangkan untuk optimasi. Untuk enkripsi gambar tidak dilakukan di ESP namun dilakukan pada server backend. Dashboard dapat diakses secara langsung di http://212.85.26.216:3000/

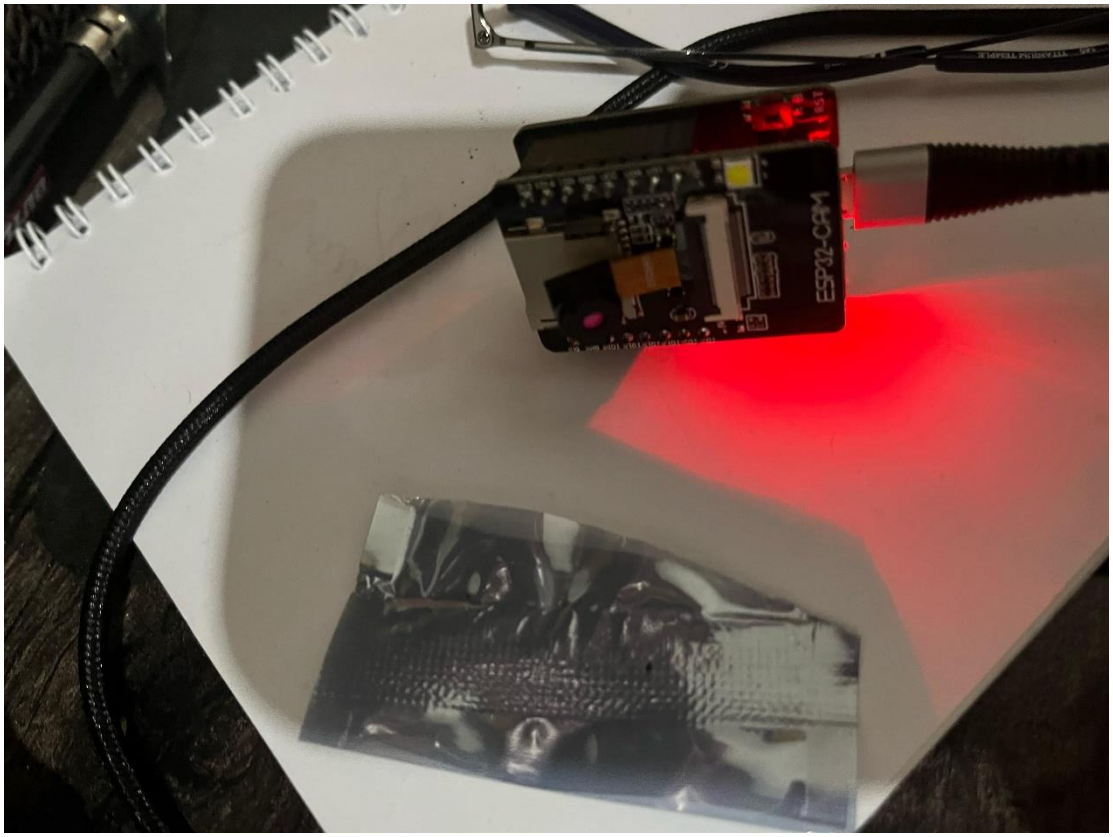
Lampiran

Image Dashboard





Dokumentasi Hardware



Data Pengujian

Data hasil pengujian 10 kali pengiriman gambar

	id	image_data	timestamp	latency_db	capture_time	publish_time
1	2000	0xED9622C92294FF315291C2770EE209A494A966F28FCEC55551C40F4…	2025-04-25 23:34:14	41.4312	0	3367
2	2001	0xDA4F440146E9AC3AE8963B88AD5B5B11C1969E81B9DB207591B26F7…	2025-04-25 23:34:21	31.086	0	729
3	2002	0x361571F8CFBDEFC850C59B8F589397D9A9CBB82D66935A1A0C6B844…	2025-04-25 23:34:31	42.8913	0	818
4	2003	0x47F397984405D27598F94D2B92BCFAE55C2329A4945DD6E4DD57FB0…	2025-04-25 23:34:40	31.0848	0	555
5	2004	0x2E2AFF1AD2E993EF64C45CF3829FE0A4FE27B328024D3B8210F952E…	2025-04-25 23:34:50	30.7369	0	280
6	2005	0x3B0454C34F3A00C664F912EDC5232A4F8B42E02268B7812729C4DF2…	2025-04-25 23:35:00	33.9532	0	276
7	2006	0x12F6E38816342F13A621319C135BBF51C66765AAB746C38581B67D9…	2025-04-25 23:35:10	49.9563	0	376
8	2007	0x0D289E1FF6B733741169E0A342463982C08B88CCCE3CA78AB521973…	2025-04-25 23:35:20	32.897	0	452
9	2008	0x4522F45314CDDD684BE0EF1282A9EDDCE6B0E3C4D033EF68A346FE9…	2025-04-25 23:35:30	32.3603	0	201
10	2009	0x89C0C872D02FC3D5C5D82978D402AD0AADF10C9C83FDBBB2D713484…	2025-04-25 23:35:40	29.6543	0	98

Data hasil pengujian 20 kali pengiriman gambar

	id	image_data	timestamp	latency_db	capture_time	publish_time
1	2051	0x848B5A7FDC581767BC519BC9BBA3EB577E8BD69FDE93FD00282BC3F…	2025-04-25 23:38:20	34.4114	0	1857
2	2052	0x99FBFA1839BA6044DFC74AE6DF4EC010A0368B282A2579B6242B138…	2025-04-25 23:38:38	61.0006	0	9316
3	2053	0xD4DE6F548B09BC71F6C6DBB250A7F51551BD6343B5A1E478E0C4A5C…	2025-04-25 23:38:43	32.7356	0	5002
4	2056	0x84A9468EE84088888DA5487760A03577AF0EC60152702E5BCEB4F2E…	2025-04-25 23:39:32	47.0665	0	4072
5	2057	0x62DB117AF5F011E616D8041FE70242EBDD6E5A4958716E823C522B8…	2025-04-25 23:39:41	31.2622	0	3331
6	2058	0x6F46DB1114F3CAE69C9CF42D008D3C4B9660D7CED7F9CC436B721FF…	2025-04-25 23:39:48	32.8822	0	341
7	2059	0x532360D5338AAA8CA13B53E34C35FDCDED0D841E31A0A2EC2C1D57C…	2025-04-25 23:39:58	34.9245	0	424
8	2060	0xF2655A187DE93E0FA980AA40A42CB531866C616E95BB19271FF7AF5…	2025-04-25 23:40:08	59.968	0	481
9	2061	0xCFAF5EF981E2FEA7C8E5F0265EAD1055F2E1846877E90983E4FC66F…	2025-04-25 23:40:18	50.3044	0	455
10	2062	0xDF854020881DFB216D00F9C98C2376135DDF5498A91A4D3EA0B7726…	2025-04-25 23:40:28	29.7644	0	304
11	2063	0xB9CAFEDBABE497344D6200C46FF5BE38ED0DAD32DB1F14822122DE3…	2025-04-25 23:40:38	32.3896	0	552
12	2064	0x2120EB2FC509F4364D426C32DD9450EB3714EBB6336876830EB8FCE…	2025-04-25 23:40:48	35.486	0	293
13	2065	0x216DA049E8F93C1F699DC17F25A94B6F6D0B1D0C59C1F0D7852FAE3…	2025-04-25 23:40:58	29.6564	0	384
14	2066	0xADC2BC55E3B367BE8588801EF324F4FDF9BE9D37BC5B04F77E1E81C…	2025-04-25 23:41:08	31.1103	0	531
15	2067	0x5077B1503B69276FAB8A2F03162239CA9FE93965CEFF5E7C1F5439C…	2025-04-25 23:41:18	36.1111	0	430
16	2068	0xA7CCC8FC38EE11ECE83866FC7F71A028B4D5E8A79A494F3E3C6CFF6…	2025-04-25 23:41:28	53.8015	0	228
17	2069	0x1AC90841824B584CA96F178287D7529315BD74AC687055AF5E75B6…	2025-04-25 23:41:38	31.112	0	350
18	2070	0x9B1D1B1232FE0E31D554314B722193D5E6422BA79F1B64C40F73197…	2025-04-25 23:41:48	36.5279	0	253

Data hasil pengujian 100 kali pengiriman gambar

	id	image_data	timestamp	latency_db	capture_time	publish_time
1	2101	0x628311692C0D85EDCCCB0AAA919BE63D51CE8CC079ABB4C02B3F34EF…	2025-04-25 23:43:27	32.5375	0	1707
2	2102	0xB1987145A87AC5A844CB833EEC02E02C95788E9597700E3FE9DF5095…	2025-04-25 23:43:46	36.9985	0	10516
3	2103	0x7F173B82852B05A5974CC7E75113A80D12C3F47374845F47578D24C0…	2025-04-25 23:43:50	30.3938	0	4052
4	2104	0x67915FCF3931DB950A0ECEB44F73237210F1FBDC5BB58B000CA1F4ED…	2025-04-25 23:43:56	32.9554	0	297
5	2105	0x5321C845A6DDD67FA2FC06ACAA5E9E5EDCC2111F1A2BD7189BE8F15C…	2025-04-25 23:44:14	47.6913	0	8146
6	2106	0xC9E1BCC3537884B96258E39B95A4ED854A09FF2AB359E72AE73A6294…	2025-04-25 23:44:21	32.7322	0	1697
7	2107	0xC99E3BA6844E62F7A8F49BADEF5442D018D0755CCDC5C6F9A5D09C7B…	2025-04-25 23:44:27	40.3926	0	973
8	2108	0x2160F53693D58B557B706D384FBF7DB0962FF06E391AAC4B199EA7F1…	2025-04-25 23:44:37	34.5666	0	1491
9	2109	0xA105D4EF49000C6D21FA908EB30D738F35DDB29CD76162AA1AC8B184…	2025-04-25 23:44:47	44.9979	0	980
10	2110	0x09FEA6F0B09BF16C36E10C838B8B21C2107592A61CC22F776447EF86…	2025-04-25 23:45:06	37.411	0	9218
11	2111	0x8C631C740922AFAFDC5F74DB9F18CBD190CCF5496BF8D48915163FB4…	<null>	305.518	0	100
12	2112	0xE136DC60E7BEE308F4DB05A497ED84D557336B847C0B7AC086E40C59…	2025-04-25 23:45:27	31.6432	0	1109
13	2113	0x24A1CCC83C550DCE71C2C5C06376042DB82566BF018A7971A0B6FAFA…	2025-04-25 23:45:34	29.223	0	338
14	2114	0x353BEFB441EE6888B83C6103F9AD55DFD060228F0EBF355B6C8B63FF…	2025-04-25 23:45:44	30.9081	0	344
15	2115	0xAA6C076B6399A3E6CA26D952BC84F8DB7F61E5B88D3939A8944AAE9B…	2025-04-25 23:45:54	34.8718	0	224
16	2116	0x4527B9758C7A7F9CF1C382B53E75E33DEAEB15432CC44E36829D030A…	2025-04-25 23:46:04	30.1135	0	306
17	2117	0x131D8931BA524464AA1EFB070D81FB20FA089E2DBB1291C61369FA92…	2025-04-25 23:46:14	57.3668	0	412
18	2118	0x53C935CCDD60B21613EA2EA7DA449A59BFCE3F20CCA7C1EEBADAACFB…	2025-04-25 23:46:24	31.9948	0	317
19	2119	0x4B6D3E25E739E1BF83744B9DE1C00AF0BFF33F33FFC2605DD63F1E07…	2025-04-25 23:46:34	32.6886	0	358
20	2120	0xBC7B496EF242DF101C4ADD38DD1AEF52E3D6AF7D54CC31E12BE2CF66…	2025-04-25 23:46:44	49.9451	0	451
21	2121	0xDE7333BC48070AA92D46FE773B5469A3DF485A5923213386005C1616…	2025-04-25 23:46:54	32.0385	0	391
22	2122	0xA99784E423AC83DAF587437A892F487230FB408E8E5011C8E7584DBC…	2025-04-25 23:47:04	55.5058	0	514
23	2123	0x569943D0EE8A77A67C9EC76FD1F48A9ABC449C434D023B23E94A7838…	2025-04-25 23:47:14	35.9175	0	366
24	2124	0xAB36DD0CD09EF8F7633D71518A8FEC976C6F739C5E9EC6E303EB26F7…	2025-04-25 23:47:24	37.2689	0	432
25	2125	0x2965814BDF557DC6A104F287A58520A27F87735D2E5EA9EC59E9C643…	2025-04-25 23:47:34	45.5763	0	289

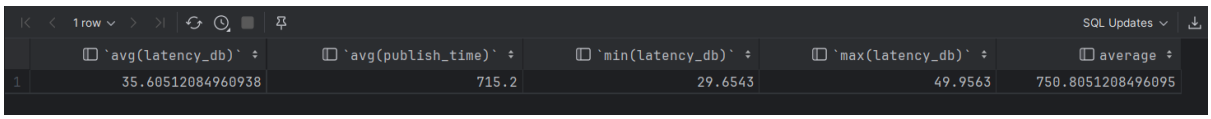
	id	image_data	timestamp	latency_db	capture_time	publish_time
26	2126	0x7865862B70E5D7BB544B305B1B54F0FFDC5E380735FD920E4201A12F…	2025-04-25 23:47:44	34.5564	0	269
27	2127	0xABACBB72CA73C3190BDAB05F05C907AB54073DE5957D817D01D7AB8D…	2025-04-25 23:47:54	33.3779	0	224
28	2128	0x1D7B89C538D9A5A172634009B5D58CAC7592E7AD163FE8BDCF32F748…	2025-04-25 23:48:04	34.4357	0	315
29	2129	0xA9EC5BB1E87FF0441131B46A4840FA08B603E24C0EB90854B0B665D0…	2025-04-25 23:48:14	31.1415	0	240
30	2130	0x5D5E65DF3D4047F85D51F563F8196F502DDA2208D64BCF8B0B4B1745…	2025-04-25 23:48:24	34.4443	0	444
31	2131	0x91CEF4D346A37B5D36D778BDD67239AAA3296B2380050CB3D95A8B73…	2025-04-25 23:48:34	52.4337	0	293
32	2132	0x532790F5CCB4509CB2CC4BA99B95262F46A04739EE3A61E473B9D2C5…	2025-04-25 23:48:44	44.8344	0	411
33	2133	0x3C0C8C195D8DE1E0F569E98100BB55EB3EBA336871ED561EC6D99830…	2025-04-25 23:48:55	34.3051	0	644
34	2134	0x08330894889F0B37DC43F42CB448B47A5A7F995B3B3C8772415ADB77…	2025-04-25 23:49:04	31.9281	0	300
35	2135	0x0C5873DD2D2B20D3959314657000855296FE666A5362851CC025891…	2025-04-25 23:49:15	33.973	0	1039
36	2136	0xA400FE05BD55FBA8014DACEE2F71A23BD9F4E8DF852D86CAB04178F2…	2025-04-25 23:49:24	31.1518	0	357
37	2137	0x5FEACB1C2E25B6A64BD5C638B8D9007A5E58CCF8A61D49279B7B9CF6…	2025-04-25 23:49:34	29.6805	0	250
38	2138	0x17646CD0EFCA1461692A2E3396D6EF4B4F3FC38228A6093B02E0B04E…	2025-04-25 23:49:44	32.9204	0	484
39	2139	0xEBA29152EB2AD47B18811CD105F27CD9C9D009F82568CB307C088950…	2025-04-25 23:49:55	30.2341	0	466
40	2140	0x6B873E8ECF2C5CF691BF30412533820BA0B7D4F1530BBCDC92E995BF…	2025-04-25 23:50:04	32.2266	0	412
41	2141	0x347C68CC7AD23BBBE66DD96C01753B2DB6F430E4628A774CCC1CEED3…	2025-04-25 23:50:14	42.9266	0	547
42	2142	0x4F3CA7F2E982B6298C1F75277F0E823B52C99E8771C4C9C6F3268D57…	2025-04-25 23:50:24	36.8524	0	505
43	2143	0x53388F86846BE0983E3788C9DC8D8500D2279C528E96F12831E80121…	2025-04-25 23:50:34	31.1549	0	461
44	2144	0x18DB351032460BD8486A4FBC38F040FAF3CC5AD764CE6A09350B24FE…	2025-04-25 23:50:44	44.5032	0	342
45	2145	0x834BD264F59CE60303EAF06E0C74C6DD565BED32C3AE0E4DC2EDED3E…	2025-04-25 23:50:54	53.7117	0	358
46	2146	0x2D14068B9DB7EDCC56E5BA3213AE87F993CAAFBE559766B41C0B21DA…	2025-04-25 23:51:04	47.1551	0	502
47	2147	0x501FAF0BA90661A4D372620DEE6ACBDB84D8B43D48A0A7CD3D5D54F2…	2025-04-25 23:51:14	35.5833	0	324
48	2148	0x81EC61A546D00A32A54281C1666CA94167AE1C02BBBC2E62E118EDC5…	2025-04-25 23:51:24	32.1834	0	277
49	2149	0xBF8BEB96661971AD60EC10DC865CD994A8F7E69B460D89D9DD83AD92…	2025-04-25 23:51:34	33.062	0	358
50	2150	0x8CB6D8FF4DE62A99B918A1792FF49C4153997E51702803FF8E605210…	2025-04-25 23:51:44	48.485	0	362

	id	image_data	timestamp	latency_db	capture_time	publish_time
51	2151	0x2270CA2B990D358EF62F96374A4F9C15590969D7CD67DC88A47A0994…	2025-04-25 23:51:55	372.923	0	463
52	2152	0xD6EB7E126BEF6FCEB4A675ADED3463F491AB947A75D2E9279AF1C299…	2025-04-25 23:52:04	33.7508	0	499
53	2153	0x0B5839395780313E2FF5866619396CD4B3EC8F857C98322F19B26644…	2025-04-25 23:52:15	31.965	0	1078
54	2154	0xE656C51695197EF28D5BA46CDE90F5047E46D0BD7C0FD48D915256B4…	2025-04-25 23:52:24	55.0778	0	240
55	2155	0xA85476826466E4231A9891D82617A0494BEEE0120187A633D242F36B…	2025-04-25 23:52:35	1124.9	0	357
56	2156	0x7789D9348346C82FD4C987E36F515F02FA7C074328A8ED36DBD23A64…	2025-04-25 23:52:44	32.2433	0	432
57	2157	0x5D43B3996248EEBAF8314BC70745913909B4669D382DBB3A943C6B57…	2025-04-25 23:52:54	39.4869	0	439
58	2158	0x5499AA074F4B92F31C2239B3DDDCF90A6BE221914CC9738B83A68125…	2025-04-25 23:53:04	46.6311	0	490
59	2159	0x27F26C2EBCAF7A448D8ECB313D6B0FDE7AF55097F6CF029900E2BEF2…	2025-04-25 23:53:14	45.4092	0	208
60	2160	0xCD04E192D47A7814719FF404CEBBA0F7F7BCFDEB04E2620AFA010EC1…	2025-04-25 23:53:24	50.6709	0	210
61	2161	0xB2F9603C02F78D7897C4E764CFC7E7F3E003BD7DA2372FD7529D6C6D…	2025-04-25 23:53:34	53.298	0	241
62	2162	0xAA118D0D7E976A1B29691002ED3D1DAACB23E4DF4C150158132411EB…	2025-04-25 23:53:45	32.228	0	534
63	2163	0x8C493FF48779F301BD65B30530C19F2C00B8DCA1C6B7CA73AADB0D52…	2025-04-25 23:53:54	32.057	0	451
64	2164	0xFB7B984B4300673FB53BF992E7837400DF1255EE6896441C52BB8609…	2025-04-25 23:54:04	41.5387	0	208
65	2165	0x9E89F12BCB7DAB3C4285AAD72DF0EEBD1CA2B7E4820FC26D21458493…	2025-04-25 23:54:14	44.2812	0	384
66	2166	0x5003A79671E394D66CA343B0329E56FE98EBB50A8EA1BBEF08B7446D…	2025-04-25 23:54:24	29.1762	0	338
67	2167	0xFFE9BFF9D5A353C3B26AB50B16FF749BF9012F00582CB3BEF1A3A3B9…	2025-04-25 23:54:34	41.4941	0	379
68	2168	0xE1D08C9330DEA1A8155E72B540D41316A4AAA57AFB5406AECBEC7A33…	2025-04-25 23:54:44	46.1349	0	334
69	2169	0xEA038EB2FCEEA280F5261DA2984C882CDC6F8651C7E48723BBCF42CF…	2025-04-25 23:54:54	37.6847	0	415
70	2170	0x6F7250334B97A6B90E5CC53E96904A9BB963FE72CD27BECAB18921D4…	2025-04-25 23:55:04	40.2083	0	340
71	2171	0x0435A4026129705330217D0FDCA6F11EB403F6EED359AC59E5ED14D5…	2025-04-25 23:55:14	43.6897	0	284
72	2172	0x709F4CE2007D7B3D471C9FCBA0C8BADBF9975CD9510820BF5F6467B2…	2025-04-25 23:55:24	40.6387	0	313
73	2173	0xA43B566FD0AB186D88C415AC5691F604DC3334D99A77403854B4AD98…	2025-04-25 23:55:34	53.1654	0	362
74	2174	0x775FFE7F7F51F1783E3AAD03DA136F9B1291FC169C4C07358114302B…	2025-04-25 23:55:46	903.734	0	695
75	2175	0x4C7A41B333BC464795BC3938A725F9B64A53E1B57548B8B77DD19AC9…	2025-04-25 23:55:57	30.3593	0	2837

76	2176	0xB1AF6AE658FFE941BE9E5F83911C36D891E01F1A8266B49060BAC501…	2025-04-25 23:56:04	38.6407	0	317
77	2177	0x7F60FD0117E0BBC5A2121D5A933B0AB7428EEF181E20A6F00D332381…	2025-04-25 23:56:14	32.3493	0	329
78	2178	0xD1DD74962B37CD85BE45365F17FA2A21A8A57CFBB25AE8ADDE2A7189…	2025-04-25 23:56:25	286.546	0	988
79	2179	0xF8947ACCB052BEF96E14DCB8272F5AF639AD9CE607CE338FA9DC0D4F…	2025-04-25 23:56:35	29.6187	0	1049
80	2180	0x2CFC140218A07AAD53926D713DE2D7126A14D89C9302235E9FD58BEA…	2025-04-25 23:56:44	33.0918	0	242
81	2181	0x284BBDD68A828D0EF90FD64114292EC9814C2DEB98271A41290507C3…	2025-04-25 23:56:55	97.3504	0	525
82	2182	0x650468623240BA521BF4D23A930EAC7726C10F4B144596A01C8951E8…	2025-04-25 23:57:04	50.4036	0	300
83	2183	0x864758632AA56A99C4E53D305D66AA87EB1C306F936BD033DD8D5654…	2025-04-25 23:57:14	52.6831	0	291
84	2184	0x609C3E576A3C5E9856A3BCBAA296B151932EFEE79E56871358B913EA…	2025-04-25 23:57:24	86.4842	0	304
85	2185	0x41BFC357CE2533258F301217F4DA130288FA99A584E29B8A1440A234…	2025-04-25 23:57:34	32.2227	0	262
86	2186	0x4A26D18E69FDB345BC4E11F398F273D0DA953054467D093E34C56604…	2025-04-25 23:57:44	35.7957	0	228
87	2187	0x57DD56F3B4FD657F68AFF59E0047563F88911268691CBB13DC393212…	2025-04-25 23:57:54	32.4125	0	290
88	2188	0xF602C265638FCDC824EF180154DB0498DD4CAE4F166315E7539EDA46…	2025-04-25 23:58:04	44.2858	0	208
89	2189	0x62A75D7623B5D873DC5ACCE64B05F78AEDF53B741FC10A668165C575…	2025-04-25 23:58:14	863.858	0	315
90	2190	0xCAC693E538CF66FD0F9784E86E7D6081D611744CE71792DD34A20406…	2025-04-25 23:58:24	31.0369	0	216
91	2191	0x96BE754C1ED405263F3C689E417F6E643BDFBCDB8BEB2F5F5793AB9F…	2025-04-25 23:58:34	30.2804	0	241
92	2192	0xC1BBF81C0E7719A45E3C0DC4D9143DFDB12958B22D530D94652CD2FE…	2025-04-25 23:58:44	54.4448	0	228
93	2193	0x940CB95807DE5AB7992EAC3771816946A331304E7F7C0744A7F8BDE1…	2025-04-25 23:58:54	32.7342	0	235
94	2194	0x6F217EE7A36FDDEE085A6B3A0FE91017003E6D8F687DC028945AD5B5…	2025-04-25 23:59:04	34.3053	0	182
95	2195	0x359F82B5985B6CC7A21C5B3D48280F98B3099A200E9B862656A6CBDF…	2025-04-25 23:59:14	46.3364	0	156
96	2196	0xBE426CF1425646FE9161B7D2573F110BC173B0BEB32AD92D8BD08D48…	2025-04-25 23:59:24	32.2948	0	255
97	2197	0x6FF541C4709B097FFFF5AEAEE19DB3F87DAACA25BFD08676435F634A…	2025-04-25 23:59:34	44.8406	0	224
98	2198	0xC8199DC317C051808EFC373887FCE50867F8C2494B5678175016F8EF…	2025-04-25 23:59:44	31.7967	0	243
99	2199	0x46C75177A6B998C405EF289DBA12F84131E33FA7B4F4D308514AB98A…	2025-04-25 23:59:54	32.4442	0	209
100	2200	0x39724D95776C72DE8F69554AE47704BD69B71B2A02E99F65573139D5…	2025-04-26 00:00:04	75.1536	0	547

Data penghitungan latency pada 10 kali pengiriman

```
select avg(latency_db), avg(publish_time), min(latency_db), max(latency_db), avg(latency_db) + avg(publish_time) as average from images where id >= 2000 and id <= 2009;
```



	`avg(latency_db)`	`avg(publish_time)`	`min(latency_db)`	`max(latency_db)`	average
1	35.60512084960938	715.2	29.6543	49.9563	750.8051208496095

Data penghitungan latency pada 20 kali pengiriman

```
select avg(latency_db), avg(publish_time), min(latency_db), max(latency_db), avg(latency_db) + avg(publish_time) as average from images where id >= 2051 and id <= 2070;
```

	`avg(latency_db)`	`avg(publish_time)`	`min(latency_db)`	`max(latency_db)`	average
1	38.91747548845079	1589.111111111111	29.6564	61.0006	1628.0285865995618

Data penghitungan latency pada 100 kali pengiriman

```
select avg(latency_db), avg(publish_time), min(latency_db), max(latency_db), avg(latency_db) + avg(publish_time) as average from images where id >= 2101 and id <= 2200;
```

	`avg(latency_db)`	`avg(publish_time)`	`min(latency_db)`	`max(latency_db)`	average
1	76.02679229736329	761.27	29.1762	1124.9	837.2967922973633

Skrip Kode

Backend

```
from flask import Flask
from flask_cors import CORS
from dotenv import load_dotenv
import os
from routes import get_images, decrypt_image_api, get_information_image
from mqtt_handler import on_connect, on_message
import paho.mqtt.client as mqtt

load_dotenv()

app = Flask(__name__)

CORS(app, resources={r"/*": {"origins": os.getenv("URL_FRONTEND")}})

app.add_url_rule('/','home', lambda: "Welcome to the API", methods=['GET'])
```

```
app.add_url_rule('/images', 'get_images', get_images, methods=['GET'])
app.add_url_rule('/images/decrypt/<int:image_id>',           'decrypt_image_api',
decrypt_image_api, methods=['GET'])
app.add_url_rule('/image/<int:image_id>','get_information_image',get_information_
image, methods=['GET'])

def setup_mqtt():
    """Fungsi untuk setup MQTT"""
    mqtt_client = mqtt.Client()
    mqtt_client.on_connect = on_connect
    mqtt_client.on_message = on_message
    mqtt_client.username_pw_set(os.getenv("MQTT_USER"), os.getenv("MQTT_PASS"))
    mqtt_client.connect(os.getenv("MQTT_BROKER"), int(os.getenv("MQTT_PORT")), 60)

    mqtt_client.loop_start()

if __name__ == "__main__":
    setup_mqtt()

    app.run(host="0.0.0.0", port=5000, debug=False)
```

Frontend

```
import './App.css';
import React from 'react';
import { BrowserRouter as Router, Route, Routes } from 'react-router-dom';
import HomePage from './page/Home';
import ImagePage from './page/Image';

function App() {
  return (
    <Router>
      <div className="container mx-auto px-4 py-8">
            <h1  className="text-4xl  font-bold  text-center  mb-6  gap-10">Image
Dashboard</h1>

        <Routes>
          <Route exact path="/" element={<HomePage />} />
          <Route path="/image/:id" element={<ImagePage />} />
        </Routes>
      </div>
    </Router>
  );
}

export default App;
```

Docker Config

```
version: '3'

services:
  backend:
    build:
      context: ./backend
    container_name: flask-backend
    ports:
      - "5000:5000"
    environment:
```

```
      - MQTT_BROKER=${MQTT_BROKER}
      - MQTT_PORT=${MQTT_PORT}
      - MQTT_USER=${MQTT_USER}
      - MQTT_PASS=${MQTT_PASS}
      - MQTT_IMAGE_TOPIC=${MQTT_IMAGE_TOPIC}
      - MQTT_LATENCY_TOPIC=${MQTT_LATENCY_TOPIC}
      - MYSQL_HOST=${MYSQL_HOST}
      - MYSQL_PORT=${MYSQL_PORT}
      - MYSQL_USER=${MYSQL_USER}
      - MYSQL_PASSWORD=${MYSQL_PASSWORD}
      - MYSQL_DB=${MYSQL_DB}
      - AES_KEY=${AES_KEY}
      - URL_FRONTEND=${URL_FRONTEND}
    networks:
      - app-network
    restart: always

  frontend:
    build:
      context: ./frontend
    container_name: react-frontend
    ports:
      - "3000:3000"
    depends_on:
      - backend
    environment:
      - REACT_APP_BACKEND_URL=${URL_BACKEND}
    networks:
      - app-network
    restart: always

networks:
  app-network:
    driver: bridge
```

Contoh Payload MQTT

Payload Image

```
{"id":4505,"image":<image_base64>,"timestamp":"2025-04-26 15:50:20"}
```

Payload Latency

```
{"id":2106,"capture_time":0,"publish_time":123}
```

Link Github : https://github.com/vierifirdaus/UTS_IOT

Link Dashboard : http://212.85.26.216:3000/