ANALISIS IMPLEMENTASI ALGORITMA ENKRIPSI AES-128 BIT DATA RFID PADA JARINGAN 802.11n UNTUK MONITORING JARAK JAUH BERBASIS MQTT

M. Apriannur[1], Dodon Turianto Nugrahadi[2], Andi Farmadi[3], Muhammad Itqan Mazdadi[4], Fatma Indriani[5]

[1,2,3,4,5]Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lambung Mangkurat
[1]aprian.muhammad09@gmail.com, [2]dodonturianto@ulm.ac.id, [3]andifarmadi@gmail.com,
[4]mazdadi@ulm.ac.id, [5]f.indriani@ulm.ac.id
Corresponding author: [2]dodonturianto@ulm.ac.id

Abstrak

MQTT merupakan protokol komunikasi *publish*/*subscribe topic-based* yang sangat sederhana dan ringan, yang didesaim untuk alat yang memiliki kemampuan terbatas, bandwidth yang rendah, latency yang tinggi atau jaringan yang kurang dapat diandalkan, sehingga implementasi berkembang pada perangkat mikrokontroller. Salah satu kekurangan MQTT yaitu secara default hanya memiliki mekanisme autentikasi saja dan belum terenkripsi. Pada penelitian ini menganalisis perbandingan kondisi RAM, *delay*, dan *throughput* dalam implementasi algoritma enkripsi AES-128bit data RFID dengan koneksi 802.11n berbasis MQTT pada perangkat TP-link Archer C54 dan TP-link TL-MR100. Skenario yang digunakan yaitu enkripsi dan tanpa enkripsi untuk pengiriman data sebanyak 50 kali. Hasil penelitian menunjukkan bahwa proses enkripsi data RFID menggunakan algoritma AES-128bit menghasilkan penggunaan RAM sebesar 23% pada perangkat 802.11n TP-Link Archer C54 dan 24% pada perangkat 802.11n TP-link TL-MR100. Proses enkripsi juga menghasilkan *delay* pengiriman data sebesar 8% pada perangkat 802.11n TP-Link Archer C54 dan 15% pada perangkat 802.11n TP-link TL-MR100. Serta proses enkripsi menghasilkan *throughput* pengiriman data sebesar 127% pada perangkat 802.11n TP-Link Archer C54 dan 117% pada perangkat 802.11n TP-link TL-MR100. Akan tetapi berdasarkan pengujian hipotesis pada 50 data, bahwa implementasi algoritma enkripsi AES-128bit data RFID dengan koneksi 802.11n berbasis MQTT memiliki hubungan terhadap penggunaan RAM, mempengaruhi *throughput* pada proses transfer data akan tetapi tidak mempengaruhi *delay*. Perbandingan penerapan pada perangkat nirkabel 802.11n bahwa TP-Link Archer C54 lebih unggul yaitu lebih rendah penggunaan RAM sebesar 1%, lebih rendah *delay* sebesar 7% dan lebih tinggi *throughput* sebesar 10% dibanding TP-link TL-MR100.

Kata kunci : *Aes-128 bit, NodeMCU ESP8266, MQTT*

1. Pendahuluan

Message Queue Telemetry Transport (MQTT) merupakan protokol pesan ringan yang dirancang untuk komunikasi mesin-ke-mesin (M2M) yang beroperasi pada lapisan atas layer 7 model OSI, juga dikenal sebagai lapisan aplikasi. Protokol MQTT, terlepas dari pemutusan, memastikan pengiriman semua pesan yang dikirim. Metode pengiriman yang digunakan oleh protokol MQTT adalah komunikasi *publish/subscribe* (Abilovani, 2018).

Protokol ini merupakan komunikasi publish/subscribe topic-based yang sangat sederhana dan ringan, yang didesaim untuk alat yang memiliki kemampuan terbatas, bandwidth yang rendah, latency yang tinggi atau jaringan yang kurang dapat diandalkan. Keunggulan utama adalah dapat mengirimkan data dengan bandwith yang ringan, konsumsi listrik yang sedikit, latensi serta konektifitas yang sangat tinggi, ketersediaan variable yang banyak serta jaminan pengiriman data yang dapat dinegosiasikan (Susanto et al., 2018). Pengimplementasian MQTT berkembang pada perangkat mikrokontroler yang merupakan sebuah komputer di dalam chip dengan spesifikasi terbatas serta yang menekankan efisiensi dan efektifitas biaya (Amarudin et al., 2020).

Protokol MQTT, dalam konfigurasi defaultnya, hanya menggabungkan mekanisme otentikasi. Terlepas dari adanya mekanisme seperti itu, individu yang tidak berwenang dapat dengan mudah memperoleh pasangan pengguna dan kata sandi tanpa adanya enkripsi, sehingga membahayakan manajemen perangkat jika administrator berusaha memodifikasi kata sandi. Proses enkripsi perlu memastikan bahwa hanya pihak yang berwenang dapat mengaksesnya (Andy & Rahardjo, 2018). Teknik kriptografi sebagai salah satu strategi yang digunakan untuk melindungi data. *Cryptosystem* berfungsi dengan

proses penyandian pesan ke dalam kode rahasia yang secara eksklusif dapat dipahami oleh aktor sistem informasi (Purwanto, 2021). Potensi pemanfaatan teknologi kriptografi, khususnya enkripsi, menghadirkan aplikasi yang layak di ranah sistem kunci pintu digital berbasis RFID. Salah satu kekurangan kunci berbasis RFID adalah kerentanannya terhadap pencurian data melalui mesin skimmer, yang memungkinkan duplikasi kunci RFID dengan mudah. Implementasi sistem enkripsi data kartu RFID dilakukan untuk mencegah pelanggaran keamanan tersebut dan memastikan perlindungan data yang disimpan pada kartu. Dengan mengintegrasikan sistem IoT (*Internet of Things*) menggunakan mikrokontroler maka sistem keamanan otomatis menjadi solusi kekurangan teknologi RFID. Selain itu juga dengan memanfaatkan protokol MQTT sebagai protokol dukungan transfer data semakin meningkatkan fungsionalitas sistem.

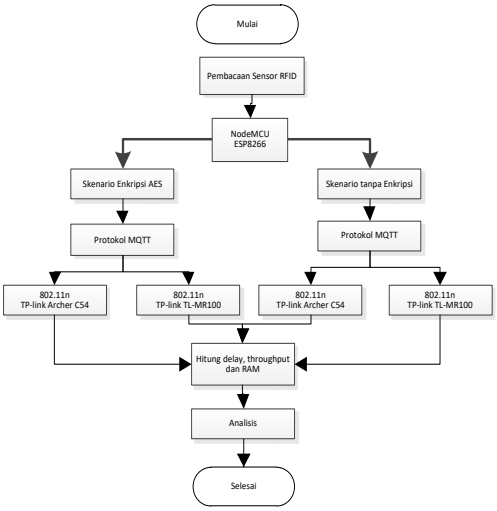
Pada metode kriptografi terdapat beberapa algoritma enkripsi, pada penelitian tentang perbandingan algoritma DES, AES, IDEA dan BLOWFISH dalam enkripsi dan deskripsi data mendapatkan hasil analisis perbandingan kecepatan dekripsi algoritma keempat algoritma di atas dapat dilihat bahwa persentasi tertinggi dari algoritma AES 45%, Blowfish 34%, DES 19% dan terakhir adalah algoritma IDEA yang hanya memiliki kecepatan 2% (Meko, 2018) (Mahajan & Sachdewa, 2013).

Proses pengamanan IoT berbasis NodeMCU menggunakan Algoritma AES pada 128 bit dapat diimplementasikan pada sistem IoT untuk mengamankan data (Zubaidi et al., 2021). Waktu yang dibutuhkan untuk enkripsi dan dekripsi file dokumen relatif AES 128 Bit lebih cepat dibanding dengan beberapa enkripsi lainnya (Prameshwari & Sastra, 2018). Sedangkan bagi mikrokontroler yang memiliki kapasitas memori yang cukup besar, hal tersebut tidak menjadi masalah. Namun jika algoritma diimplementasikan pada mikrokontroler dengan kapasitas memori yang kecil, maka akan menjadi masalah (Rizqulloh et al., 2021). Aspek kritis dari sebuah sistem berbasis IoT adalah kualitas layanan jaringan. Kualitas layanan jaringan sangat menentukan ketersediaan dan kecepatan konektivitas antar perangkat IoT (Enda dkk, 2021).

Berdasarkan keterbatasan MQTT dalam proses pengamanan tersebut, maka diperlukan penerapan enksipsi. Salah satu dengan tingkat pengamanan tinggi, memiliki kecepatan enkripsi dan dekripsi yaitu AES. Pada penelitian ini menggunakan AES 128bit pada jaringan 802.11n dengan membandingkan hasil parameter kualitas layanan yaitu penggunaan memory (RAM), *delay* dan *throughput* pada 2 perangkat nirkabel 802.11n yaitu TP-link Archer C54 dan TP-link TL-MR100.

2. Metode Penelitian

Adapun alur penelitian yang dilaksanakan dalam penelitian ini dapat dilihat pada Gambar 1:



Gambar 1. Alur penelitian

Pada diagram ini menunjukan skenario penelitian yang dilakukan adalah dimulai dengan pembacaan sensor RFID dan dilakukan dengan mikrokontroler NodeMCU ESP80266. Selanjutnya dilakukan pengiriman data, dengan enkripsi AES 128 dan tanpa enkripsi yang dibungkus dalam protokol MQTT melalui koneksi 802.11n. Koneksi 802.11n melalui dua perangkat nirkabel, TP-link Archer C54 dan TP-link TL-MR100. Hasil dari masing-masing skenario yaitu nilai parameter penggunaan memory (RAM), *delay* dan *throughput* sebagai bahan analisis *Quality of Service* (QoS).

2.1 Tujuan Penelitian

Tujuan penelitian ini adalah untuk mengetahui pengaruh proses enkripsi AES 128-bit data RFID pada mikrokontroler NodeMCU dengan protokol MQTT menggunakan jaringan 802.11n Archer dan jaringan 802.11n MR100 terhadap penggunaan RAM, *delay*, dan *throughput*.

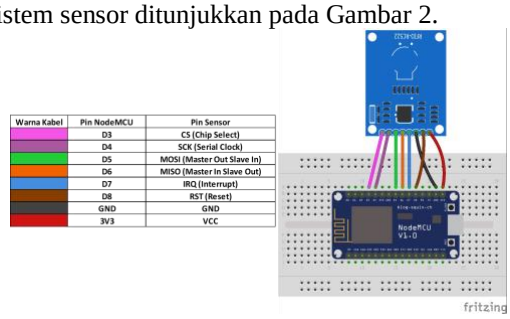
2.2 Perancangan Hardware

Pada penelitian, hardware yang digunakan adalah mikrokontroler NodeMCU ESP8266, adaptor 5V, dan sensor MFRC522 yang diimplementasikan pada perangkat TP-link Archer C54 dan TP-link TL-MR100.

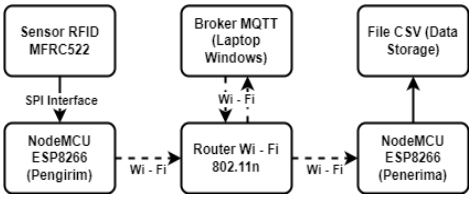
NodeMCU ESP8266 adalah *Wifi Serial Transceiver Module* sebuah komponen chip terintegrasi yang didesain untuk keperluan dunia masa kini yang serba tersambung (Roihan et al., 2016). NodeMCU ESP8266 juga menggunakan System on Chip dengan stack protokolTCP/IP yang telah terintegrasi atau saling terhubung, sehingga mudah diakses menggunakan mikrokontroler melalui komunikasi serial 802.11/g/n Wi-Fi Direct

(P2P) (Wiranto dkk, 2020).

RFID yang merupakan singkatan dari Radio Frequency Identification yang merupakan teknologi identifikasi baru yang dalam pengoperasiannya terjadi kontak antara transponder atau pembawa data (Oktaviani et al., 2020). Mikrokontroler dan sensor dihubungkan dengan menggunakan antarmuka SPI (*Serial Peripheral Interface*) dengan memanfaatkan pin yang ada pada mikrokontroler. Sensor dihubungkan dengan bagian pengirim (*publisher* MQTT) sedangkan bagian penerima (*subscriber* MQTT) tidak dihubungkan dengan komponen apapun karena fungsinya hanya menerima dan menyimpan data. Data sementara disimpan pada *chip flash memory* yang tertanam pada mikrokontroler dan dapat diakses dengan menggunakan aplikasi Thonny. Skema rancangan sistem sensor ditunjukkan pada Gambar 2.



Gambar 2. Skematik Rancangan Sistem Sensor

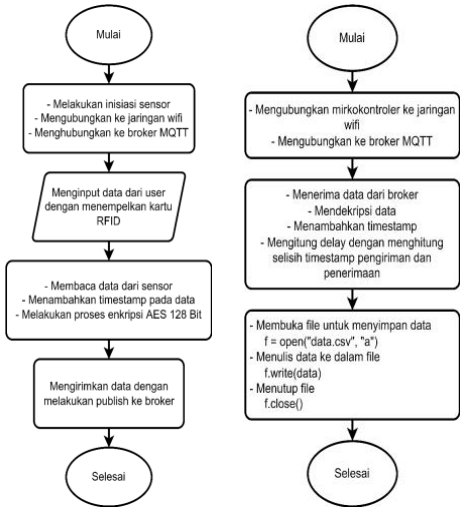


Gambar 3. Rancangan Skenario Pengujian perangkat 802.11n

Pada gambar 3, merupakan skenario pengujian koneksi 802.11n. Sensor membaca data RFID pada perangkat mikrokontroler NodeMCU ESP80266 sebagai node pengirim dan melakukan mengkonversi data dari binary menjadi string untuk proses enkripsi atau tanpa enkripsi. Data yang telah diproses kemudian dikirimkan ke broker MQTT melalui koneksi 802.11n pada perangkat TP-link Archer C54 atau TP-link TL-MR100. Selanjutnya, data yang telah diterima oleh broker akan dikirimkan ke NodeMCU ESP80266 sebagai node penerima dengan protokol MQTT koneksi 802.11n pada perangkat TP-link Archer C54 atau TP-link TL-MR100. Data yang telah diterima menyimpan hasilnya ke dalam file berekstensi CSV.

2.3 Perancangan Software

Perangkat lunak yang digunakan dalam penelitian ini dibagi menjadi dua bagian yaitu perangkat lunak pengirim dan perangkat lunak penerima. Perangkat pengirim membutuhkan lebih banyak proses karena harus membaca data dari sensor RFID, memproses data sebelum mengirimnya, mengenkripsi data dan mengirimkannya ke *broker* MQTT. Di sisi lain, fungsi penerima hanya untuk menerima data, mendekodekannya, dan menyimpan data tersebut.



Gambar 4. Diagram proses pengirim dan penerima MQTT

Pada gambar 4, menunjukkan *flowchart* untuk proses pengiriman data (*flowchart* sebelah kiri). Proses dimulai dengan inisiasi sensor. Node pengirim melakukan proses konektivitas dengan koneksi 802.11 dan MQTT *broker*. Kemudian proses pembacaan data RFID dari kartu RFID, menambahkan penanda waktu kirim dan proses enkripsi AES 128 bit. Setelah data terenkripsi maka dilakukan pengiriman data ke MQTT *Broker* melalui koneksi 802.11n.

Proses penerima MQTT (*flowchart* sebelah kanan), dimulai proses konektivitas node penerima dengan koneksi 802.11 dan MQTT *broker*. Selanjutnya menerima data, melakukan dekripsi dan menambahkan penanda waktu terima untuk disimpan dalam file berekstensi CSV.

2.4 Alur Enkripsi dan Deskripsi

Advanced Encryption Standard (AES) merupakan algoritma enkripsi dengan algoritma enkripsi kunci simetris standar. Algoritma AES, algoritma simetris yang menggunakan kunci yang sama untuk proses enkripsi dan dekripsinya (Muharram et al., 2018). Pada proses enkripsi AES terdiri dari empat jenis transformasi termasuk SubBytes, ShiftRows, MixColumns, AddRoundKey, tetapi tidak ada transformasi Mixcolumns yang dilakukan pada putaran terakhir. (Darwes et al., 2018). Proses enkripsi dan dekripsi

dilakukan oleh pengirim dan penerima masing-masing. Proses enkripsi dan dekripsi membutuhkan beberapa langkah.

1. Dimulai fungsi AES.
2. String data input dengan menempelkan kartu RFID ke sensor.
3. Data string yang diterima dan diubah menjadi bentuk biner agar proses enkripsi AES dapat dilakukan.
4. Data enkripsi tipe binary dan dikirim ke broker melalui protokol MQTT.

2.5 MQTT

MQTT adalah singkatan dari *Message, Queue Telemetry Transport*. Ini adalah protokol komunikasi *publish/subscribe* berbasis topik yang sangat sederhana dan ringan yang dirancang untuk perangkat dengan fungsionalitas terbatas, bandwidth rendah, latensi tinggi, atau jaringan yang tidak dapat diandalkan (Sutomo dan Saputri, 2018). Proses dekripsi mirip dengan proses enkripsi dalam urutan terbalik. Data yang diterima dari broker tetap berupa dekripsi biner menggunakan AES. Data yang didekripsi tetap dalam format biner, tetapi diubah menjadi string untuk disimpan lebih lanjut dalam file CSV.

MQTT awalnya dikembangkan pada tahun 1999 oleh Dr. Andy Standford-Clark dari IBM dan Arlen Nipper dari Arcom (sekarang Eurotech). Protokol ini dirancang untuk memiliki *overhead* rendah, dengan mempertimbangkan persyaratan bandwidth dan keterbatasan komputasi CPU. Spesifikasi MQTT 3.1.1 telah distandarisasi oleh konsorsium OASIS, dan tersedia dalam berbagai format. Sejak 2016, MQTT telah memanfaatkan standar ISO (IEC 20922) (Atmoko, 2019). MQTT menawarkan keuntungan dari mentransmisikan data dengan bandwidth minimal, konsumsi listrik rendah, latensi dan konektivitas tinggi, ketersediaan berbagai variabel, dan pengiriman data yang dapat dinegosiasikan yang dijamin (Susanto et al., 2018).

2.6 Throughput dan delay

Pada penelitian *throughput* merupakan kinerja jaringan yang terukur pada waktu tertentu (Wulandari, 2017). Jadi dapat disimpulkan *throughput* merupakan paket yang berhasil diterima selama interval waktu tertentu. Terdapat rumus umum yang digunakan untuk mengetahui nilai *throughput*.

$$Throughput = \frac{Packet\ data\ diterima}{Lama\ pengamatan}$$

Sedangkan *delay* merupakan waktu yang diperlukan data untuk menempuh jarak dari asal paket dikirimkan hingga sampai ke tujuan. Faktor-faktor yang mempengaharuhi *delay* adalah dapat jarak, media fisik, kongesti atau juga waktu proses yang lama.

$$Delay = \frac{Total\ waktu\ paket\ diterima}{Total\ paket\ diterima}$$

2.7 Uji Hipotesis

Pada penelitian (Ghozali, 2016) Uji hipotesis merupakan pengujian yang menguji seberapa berpengaruh variabel X dan variabel Y yang hendak di uji. Uji hipotesis merupakan pengujian khusus yang dilakukan untuk analisis regresi.
Cara menentukan ttabel, sebagai berikut:

$$T_{tabel} = (\ ^{Sig}/_{2,\ N-K})$$

Keterangan:
Sig/2 = nilai probabilita
N = Jumlah sampel
K = Jumlah variabel X & Y.

Hipotesis pengujiannya pada parameter penggunaan RAM yaitu,
H0: Tidak ada pengaruh penggunaan RAM pada transfer data menggunakan enkripsi dan tanpa enkripsi.
H1: Ada pengaruh antara penggunaan RAM pada transfer data menggunakan enkripsi dan tanpa enkripsi.

Hipotesis pengujiannya pada parameter delay yaitu,
H0: Tidak ada pengaruh delay pada transfer data menggunakan enkripsi dan tanpa enkripsi.
H1: Ada pengaruh antara delay pada transfer data menggunakan enkripsi dan tanpa enkripsi.

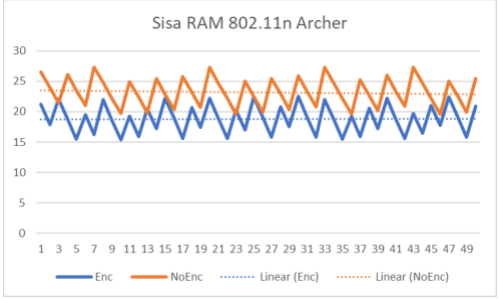
Hipotesis pengujiannya pada parameter *throughput* yaitu,
Rumusan hipotesis:
H0: Tidak ada pengaruh *throughput* pada transfer data menggunakan enkripsi dan tanpa enkripsi.
H1: Ada pengaruh antara *throughput* pada transfer data menggunakan enkripsi dan tanpa enkripsi.

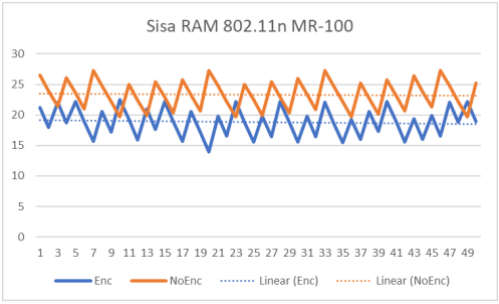
3. Hasil dan Pembahasan

Data hasil pengujian yang digunakan berjumlah masing-masing 50 data berdasarkan skenario dengan enkripsi dan tanpa enkripsi, kemudian menganalisis jumlah RAM yang tersisa, jumlah *delay* dan jumlah *throughput*.

1. Data Penggunaan RAM Dengan Perangkat 802.11n Archer dan 802.11n MR100

Hasil pengambilan data memberikan pergerakan penggunaan RAM seperti pada grafik gambar 5.





Gambar 5. Grafik trend sisa RAM

Tabel 1. Hasil Uji Hipotesis perangkat 802.11n Archer dan 802.11n MR100

802.11n Archer

Model		Unstandardized Coefficients		Standardized Coefficients Beta	t	Sig.
		B	Std. Error			
1	(Constant)	23.225	.338		68.647	.000
	Enkripsi (X)	-4.395	.478	-.680	-9.186	.000

a. Dependent Variable : Sisa RAM (Y)

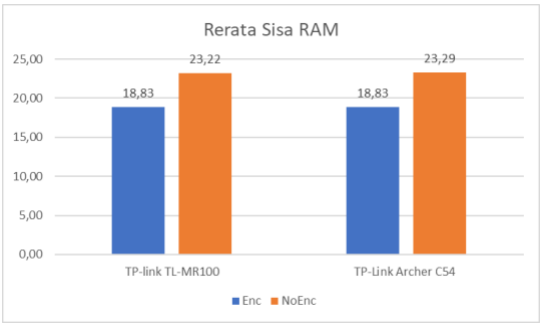
802.11n MR100

Model		Unstandardized Coefficients		Standardized Coefficients Beta	t	Sig.
		B	Std. Error			
1	(Constant)	23.287	.340		68.564	.000
	Enkripsi (X)	-4.460	.480	-.684	-9.285	.000

a. Dependent Variable : Sisa RAM (Y)

Nilai $t_{hitung} < t_{tabel}$ sehingga H0 ditolak dan H1 diterima.

Maka diketahui terdapat hubungan antara penggunaan RAM terhadap enkripsi yang digunakan (dengan enkripsi atau tanpa enkripsi). Grafik perbedaan rata-rata sisa RAM dengan kapasitas 32 KB pada mikrokontroler penggunaan transfer data melalui MQTT dengan dan tanpa enkripsi ditunjukkan oleh gambar 6.

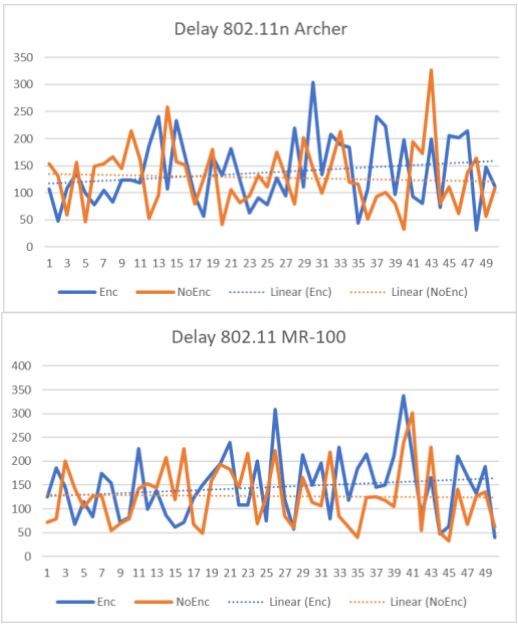


Gambar 6. Grafik rata-rata penggunaan RAM dengan 802.11n Archer dan 802.11n MR100

Pada grafik gambar 6 terlihat bahwa transfer data menggunakan enkripsi menyisakan lebih sedikit RAM. Dengan memperhitungkan kapasitas RAM dari mikrokontroler NodeMCU ESP8266 yang sebesar 32 KB. Pada tipe 802.11n Archer, skenario dengan enkripsi diperoleh hasil sisa RAM rata – rata sebesar 58,84%. Sedangkan pada skenario tanpa enkripsi, sisa RAM rata – rata sebesar 72,58%. Pada tipe perangkat 802.11n MR100, skenario dengan enkripsi diperoleh hasil sisa RAM rata – rata sebesar 58,84%. Sedangkan pada skenario tanpa enkripsi, sisa RAM rata – rata sebesar 72,77%.

2. Data penggunaan *delay* dengan Perangkat 802.11n Archer dan 802.11n MR100

Hasil pengambilan data memberikan pergerakan *delay* seperti pada grafik gambar 7.



Gambar 7. Grafik trend *delay*

Tabel 2. Hasil Uji Hipotesis *Delay* perangkat 802.11n Archer dan 802.11n MR100

802.11n Archer

Model		Unstandardized Coefficients		Standardized Coefficients Beta	t	Sig.
		B	Std. Error			
1	(Constant)	128.479	8.609		14.924	.000
	Enkripsi (X)	9.309	12.113	.078	.768	.444

a. Dependent Variable : Delay (Y)

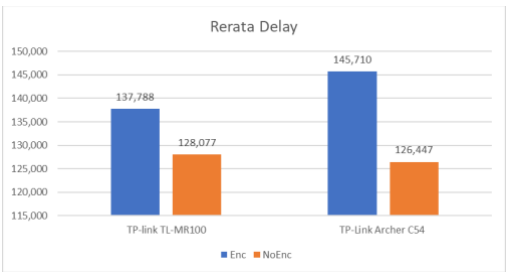
802.11n MR100

Model		Unstandardized Coefficients		Standardized Coefficients Beta	t	Sig.
		B	Std. Error			
1	(Constant)	126.447	9.118		13.868	.000
	Enkripsi (X)	19.263	12.895	.149	1.494	.138

a. Dependent Variable : Delay (Y)

Nilai $t_{hitung} < t_{tabel}$ sehingga H0 diterima dan H1 ditolak

Maka diketahui penggunaan enkripsi pada transfer data menggunakan MQTT (dengan enkripsi atau tanpa enkripsi) tidak mempengaruhi nilai *delay* pada proses transfer data. Grafik perbedaan *delay* antara transfer data melalui MQTT dengan dan tanpa enkripsi ditunjukkan oleh gambar 8.



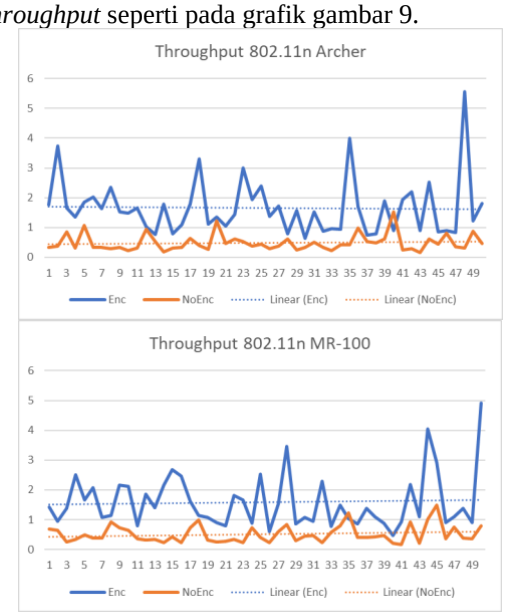
Gambar 8. Grafik rata-rata penggunaan *delay* dengan 802.11n Archer dan 802.11n MR100

Pada grafik pada gambar 8, nilai *delay* rata-rata pada transfer data menggunakan enkripsi cenderung lebih tinggi dibandingkan tanpa enkripsi. Hal tersebut karena proses pengiriman menggunakan enkripsi memerlukan waktu lebih lama untuk pemrosesan data dibandingkan dengan pengiriman tanpa enkripsi. Pada tipe perangkat 802.11n Archer, Transfer data menggunakan enkripsi memiliki *delay* rata – rata sebesar 137,8 ms. Sedangkan transfer data tanpa menggunakan enkripsi memiliki *delay* rata – rata sebesar 128,1 ms. Pada tipe perangkat 802.11n MR100, Transfer data menggunakan enkripsi memiliki *delay* rata – rata sebesar 145,7 ms. Sedangkan transfer data tanpa menggunakan enkripsi memiliki *delay* rata – rata sebesar 126,4 ms.

Hal demikian terjadi karena mikrokontroler NodeMCU ESP8266 hanya mendukung jaringan nirkabel 802.11 dengan frekuensi 802.11n, sehingga mikrokontroler hanya dapat beroperasi pada frekuensi 802.11n meskipun terhubung ke perangkat yang mendukung jaringan nirkabel 802.11ac.

3. Data Penggunaan *throughput* Perangkat 802.11n Archer dan 802.11n MR100

Hasil pengambilan data memberikan pergerakan *throughput* seperti pada grafik gambar 9.



Gambar 9. Grafik trend *throughput*

Tabel 3 Hasil Uji Hipotesis *throughput* 802.11n Archer dan 802.11n MR100

802.11n Archer

Model		Unstandardized Coefficients		Standardized Coefficients Beta	t	Sig.
		B	Std. Error			
1	(Constant)	.492	.098		4.993	.000
	Enkripsi (X)	1.168	.139	.646	8.385	.000

a. Dependent Variable : Throughput (Y)

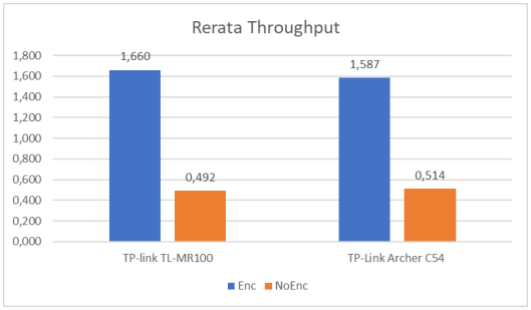
802.11n MR100

Model		Unstandardized Coefficients		Standardized Coefficients Beta	t	Sig.
		B	Std. Error			
1	(Constant)	.514	.094		5.449	.000
	Enkripsi (X)	1.073	.133	.631	8.049	.000

a. Dependent Variable : Throughput (Y)

Nilai $t_{hitung} < t_{tabel}$ sehingga H0 ditolak dan H1 diterima.

Maka diketahui penggunaan enkripsi pada transfer data menggunakan MQTT (dengan enkripsi atau tanpa enkripsi) mempengaruhi nilai *throughput* pada proses transfer data. Grafik perbedaan *throughput* antara transfer data melalui MQTT dengan dan tanpa enkripsi ditunjukkan oleh gambar 10.



Gambar 10. Grafik rata-rata penggunaan *throughput* 802.11n Archer dan 802.11n MR100

Hasil yang ditunjukkan oleh grafik pada gambar 10, bahwa terdapat perbedaan yang signifikan pada *throughput* transfer data menggunakan enkripsi memberikan hasil *throughput* yang lebih besar dibandingkan dengan transfer data tanpa enkripsi. Pada perangkat 802.11n Archer, skenario dengan enkripsi diperoleh *throughput* rata – rata sebesar 1,587 KB/s. Sedangkan pada skenario tanpa enkripsi, diperoleh *throughput* rata – rata sebesar 0,514 KB/s. Pada tipe perangkat 802.11n MR100, skenario dengan enkripsi diperoleh *throughput* rata – rata sebesar 1,660 KB/s. Sedangkan pada skenario tanpa enkripsi, diperoleh *throughput* rata – rata sebesar 0,492 KB/s.

Hal ini terjadi dengan enkripsi, panjang string setelah proses enkripsi bertambah sekitar 4 kali lipat. Selain itu, proses enkripsi juga menambahkan padding berupa random string untuk meningkatkan keamanan enkripsi dan mencegah *brute force attack* sehingga panjang akhir string setelah dilakukan proses enkripsi menjadi lebih panjang.

Berdasarkan pengujian yang telah dilakukan. Proses enkripsi AES 128 bit dalam transfer data sensor RFID menggunakan protokol MQTT pada koneksi 802.11 mempengaruhi penggunaan RAM mikrokontroler NodeMCU ESP8266. Demikian juga dengan *throughput*, secara signifikan berpengaruh karena proses enkripsi. Akan tetapi pada *delay* tidak menunjukkan adanya pengaruh enkripsi terhadap *delay*.

4. Kesimpulan

Berdasarkan hasil pengujian dan analisis yang telah dilakukan, menunjukkan bahwa proses enkripsi data RFID menggunakan algoritma AES-128 bit dengan prokotol MQTT pada koneksi 802.11 memiliki hubungan dengan parameter penggunaan RAM dan parameter *throughput*, akan tetapi tidak dengan parameter *delay*.

Walaupun secara persentase, terjadi peningkatan penggunaan RAM sebesar 23% pada perangkat 802.11n Archer dan 24% pada perangkat 802.11n MR100. Proses enkripsi juga meningkatkan *delay* pengiriman data sebesar 8% pada perangkat 802.11n Archer dan 15% pada perangkat 802.11n MR100. Serta proses enkripsi meningkatkan *throughput* pengiriman data sebesar 127% pada perangkat 802.11n Archer dan 117% pada perangkat 802.11n MR100. Sehingga perbandingan penerapan pada perangkat nirkabel 802.11n bahwa TP-Link Archer C54 lebih unggul yaitu lebih rendah penggunaan RAM sebesar 1%, lebih rendah *delay* sebesar 7% dan lebih tinggi *throughput* sebesar 10% dibanding TP-link TL-MR100.

Penelitian kedepannya, perlakukan eksperimen dan observasi dengan menggunakan protokol jaringan 802.11 ac dan ax, ataupun menggunakan protokol enkripsi lain serta penerapan pada 192 bit dan 256 bit.

DAFTAR PUSTAKA :

Abilovani, Z. B. (2018). *Implementasi Protokol MQTT Untuk Sistem Monitoring Perangkat IoT*. Universitas Brawijaya.

Amarudin, A., Saputra, D. A., & Rubiyah, R. (2020). Rancang Bangun Alat Pemberi Pakan Ikan Menggunakan Mikrokontroler. *Jurnal Ilmiah Mahasiswa Kendali Dan Listrik*, 1(1), 7–13.

Andy, S., & Rahardjo, B. (2018). Keamanan Komunikasi Pada Protokol MQTT untuk Perangkat IoT. *Prosiding-Seminar Nasional Teknik Elektro UIN Sunan Gunung Djati Bandung*, 176–184.

Darwis, D., Prabowo, R., & Hotimah, N. (2018). Kombinasi Gifshuffle, Enkripsi AES dan Kompresi Data Huffman Untuk Meningkatkan Keamanan Data. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIIK)*, 5(4), 389–394.

Enda, D. ., Subandri, M. A. ., & Supria. (2021). Analisis QoS (Quality Of Service) Sistem Monitoring Pintar Mitigasi Penularan Covid-19 Berbasis IoT. *Jurnal Informatika Polinema*, *8*(1), 39–46. https://doi.org/10.33795/jip.v8i1.705

Ghozali, Imam. (2018). *Aplikasi Analisis Multivariate Dengan Program IBM SPSS 25*. Semarang : Badan Penerbit-UNDIP.

Mahajan, P., Sachdeva, A. (2013). A Study of Enryption Algorithms AES, DES and RSA for Security. Global Journal of Computer Science and Technology, Network Web & Security, 13. Diakses https://globaljournals.org/GJCST_Volume13/4-A-Study-ofEncryption-Algorithms.pdf

Meko, D. A. (2018). Perbandingan Algoritma DES, AES, IDEA Dan Blowfish dalam Enkripsi dan Dekripsi Data. *Jurnal Teknologi Terpadu (JTT)*, 4(1).

Muharram, F., Aziz, H., & Manga, A. R. (2018). Analisis Algoritma pada Proses Enkripsi dan Dekripsi File Menggunakan Advanced Encryption Standard (AES). *Prosiding SAKTI (Seminar Ilmu Komputer Dan Teknologi Informasi)*, 3(2), 112–115.

Oktaviani, R., Nazwirman, N., Djamaludin, D., & Windyasari, V. S. (2020). Aplikasi Sistem Parkir Kendaraan Bermotor Menggunakan Teknologi Radio Frequency Identification (RFID) Di Universitas Islam Syekh Yusuf Tangerang. *Jurnal Ilmiah Fakultas Teknik*, 1(2), 96–103.

Prameshwari, A., & Sastra, N. P. (2018). Implementasi Algoritma Advanced Encryption Standard (AES) 128 Untuk Enkripsi dan Dekripsi File Dokumen. *Jurnal Eksplora Informatika*, *8*(1), 52–58.

Purwanto, H. (2021). Penerapan Keamanan Basis Data Dengan Teknik Enkripsi. *JSI (Jurnal Sistem Informasi) Universitas Suryadarma*, *1*(1).

Rizqulloh, M. A., Somantri, Y., Pramudita, R., & Ramelan, A. (2021). Implementasi algoritma block cipher four pada mikrokontroler STM32F103C8T6: Implementation of block cipher four algorithm on STM32F103C8T6 microcontroller. *JITEL (Jurnal Ilmiah Telekomunikasi, Elektronika, Dan Listrik Tenaga)*, 1(2), 175–188.

Roihan, A., Permana, A., & Mila, D. (2016). Monitoring kebocoran gas menggunakan mikrokontroler arduino uno dan esp8266 berbasis internet of things. *Innovative Creative and Information Technology*, 2(2), 170–183.

Susanto, B. M., Atmadji, E. S. J., & Brenkman, W. L. (2018). Implementasi Mqtt Protocol Pada Smart Home Security Berbasis Web. Jurnal Informatika Polinema, 4(3), 201.

Sutomo, B., & Saputri, T. A. (2018). Remote Home Monitoring Menggunakan Protokol MQTT. *Prosiding Seminar Nasional Darmajaya*, 1(1), 146–153.

Wiranto, H. ., Solehudin, A. ., & Irawan, A. S. Y. . (2020). Utilization of NodeMCU ESP8266 and RFID Technology as Recognition of Teacher Honors. *Jurnal Informatika*

Polinema, 7(1), 17–22. https://doi.org/10.33795/jip.v7i1.404

Zein, S., L, Yasfira., R, Ghozi., E, Harahap., FH, Badruzzaman., & D, Darmawan (2019). Pengolahan Dan Analisis Data Kuantitatif Menggunakan Aplikasi SPSS. *Jurnal Teknologi Pendidikan dan Pembelajaran*,1,4.

Zubaidi, A., Sardi, R. I., & Jatmika, A. H. (2021). Pengamanan Internet of Things Berbasis NodeMCU Menggunakan Algoritma AES pada Arsitektur Web Service REST. *Edumatic: Jurnal Pendidikan Informatika*, *5*(2), 252–260.