# INTERNET OF THINGS (IoT) – TANTANGAN DAN KEAMANAN IOT MENGGUNAKAN ENKRIPSI AES

Yulita Fatma Andriani[1)], Muhammad Fajrian Noor[2)], Abidullah S. Salim[3)], Hanafi[4)]

[1, 2, 3, 4]Megister Teknik Informatika Universitas Amikom Yogyakarta
Email : [1]yulita.andriani@students.amikom.ac.id, [2]mfajrian09@gmail.com,
[3]Abysalim007@gmail.com, [4]hanafi@amikom.ac.id

**Abstrak**

Perkembangan Internet of Things (IoT) begitu pesat sehingga memberikan pengaruh baik secara individu maupun secara kelompok, baik pelaku bisnis maupun pekerja profesional. Dari perkembangan IoT yang begitu pesat tersebut terdapat beberapa tantangan penelitian dan keamanan pada perangkat IoT. Dari beberapa literatur tentang keamanan jaringan yang penulis baca, didalam paper ini penulis akan memaparkan tantangan apa saja yang harus dihadapi dalam pengembangan IoT baik dibidang penelitiannya dan bagaimana proses enkripsi Advanced Encryption Standard (AES) berjalan. Karena isu tentang keamanan terutama di dalam perangkat IoT sangatlah penting untuk didiskusikan maupun diteliti terlebih dahulu sebelum pengimplementasian Internet of Things (IoT) itu sendiri.

**Kata kunci:** Aes, IoT, Keamanan

## 1. PENDAHULUAN

Ketika istilah *Internet of Things* (IoT) adalah yang pertama diperkenalkan, pertanyaan awal bisa apa yang dianggap sebagai objek fisik. Hingga beberapa tahun terakhir, kelompok peneliti dan organisasi mencoba mengusulkan definisi IoT dengan dunia di mana objek fisik diintegrasikan ke dalam jaringan internet.

*Internet of Things*(IoT) secara umum memiliki konsep kumpulan dari banyak objek, layanan, manusia, dan perangkat yang saling berhubungan yang dapat berkomunikasi, berbagi data, dan informasi untuk mencapai tujuan bersama di berbagai bidang dan aplikasi. IoT memiliki banyak domain implementasi seperti transportasi, pertanian, perawatan kesehatan, produksi dan distribusi energi. Perangkat di IoT mengikuti pendekatan Manajemen Identitas untuk diidentifikasi dalam kumpulan perangkat yang serupa dan heterogen. Demikian pula, suatu wilayah di IoT dapat didefinisikan oleh alamat IP tetapi dalam setiap wilayah masing-masing entitas memiliki keunikan. IoT dalam berbagai bentuknya telah mulai diaplikasikan pada banyak aspek kehidupan manusia.

Tujuan IOT adalah untuk mengubah cara kita hidup hari ini dengan membuat perangkat cerdas di sekitar kita melakukan tugas dan tugas sehari-hari. Rumah pintar, kota pintar, transportasi pintar dan infrastruktur, dll. Adalah istilah yang digunakan sesuai dengan IoT. Ada banyak domain aplikasi IoT, mulai dari lingkungan pribadi hingga perusahaan (Mahmoud dkk. 2015).

Begitu cepatnya perkembangan berbagai teknologi IoT, membuat kehidupan manusia menjadi jauh lebih nyaman. Dari sisi pengguna masing-masing orang, IoT sangat terasa pengaruhnya dalam kehidupan masing-masing orang seperti pada aplikasi rumah dan mobil cerdas. Dari sisi penguna bisnis, IoT sangat berpengaruh dalam meningkatkan jumlah produksi serta kualitas produksi, mengawasi distribusi barang, mempersingkat waktu ketidaktersedian barang pada pasar retail, mencegah pemalsuan, manajemen rantai pasok, dan lain sebagainya (Meutia 2015).

Pekembangan IoT beberapa tahun ini juga karena adanya teknologi Radio Frequency Identification (RFID) dan Wireless Sensor Networks (WSN). RFID sebuah teknologi yang menggunakan komunikasi via gelombang elektromagnetik untuk merubah data antara terminal dengan suatu objek seperti produk barang, hewan, ataupun manusia dengan tujuan untuk identifikasi dan penelusuran jejak melalui penggunaan suatu piranti yang bernama RFID tag. Dan WSN

adalah sebuah jaringan yang menghubungkan perangkat-perangkat seperti sensor node, router dan sink node.

Tantangan yang ada dengan adanya teknologi seperti RFID dan WSN adalah tantangan keamanan. Bagaimana pengamanan yang dapat melindungi setiap bagian sistem dari ancamanancaman. Secara garis besar, ada tiga hal dari IoT yang dapat diancam keamanannya. Yang pertama adalah keamanan fisik, terutama keamanan sensor dan RFID dari intereferensi, dan pencegatan sinyal. Kedua adalah keamanan operasi pada berbagai elemen yang harus dapat menjamin bahwa sensor, sistem transmisi dan lainnya dapat beroperasi secara normal. Keamanan operasi ini pada dasarnya sama dengan keamanan sistem informasi tradisional. Terakhir adalah keamanan data, yang juga meliputi berbagai elemen. Informasi pada sensor, sistem transmisi dan pengolah data tidak boleh di rusak, dicuri maupun dipalsukan. Selain ketiga hal di atas, jaringan sensor juga menghadapi persoalan keterbatasan daya. Karena itu, selain menghadapi persoalan keamanan jaringan, IoT juga diancam oleh serangan dan ancaman yang spesifik bagi IoT(Meutia 2015). Salah satu enkripsi yang dapat dapat mengamankan data yang ada di IoT adalah *Advanced Encryption Standard* (AES).

Dari semua yang sudah dijelaskan di atas di dalam paper ini akan memaparkan tantangan apa saja yang harus dihadapi dalam pengembangan IoT baik dibidang penelitiannya dan bagaimana proses enkripsi AES berjalan.

## 2. TINJAUAN PUSTAKA

### a. Arsitektur IoT

*Internet of Things*(IoT) menurut banyak peneliti memilik tiga lapisan yang disebut *perception layer, network layer,* dan *application layer*. Setiap lapisan ditentukan oleh fungsinya dan perangkat yang digunakan di lapisan itu. Berikut adalah penjabaran masing-masing lapisan yang ada di IoT(Mahmoud dkk. 2015).

### b. *Perception layer*

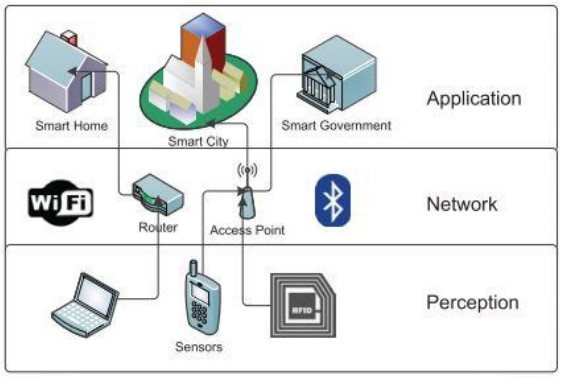
Pada lapisan ini juga yang kita kenal sebagai sensor. Tujuan dari lapisan ini adalah untuk memperoleh data dari lingkungan dengan bantuan sensor dan aktuator. Lapisan ini mendeteksi, mengumpulkan, dan memproses informasi dan kemudian mengirimkannya ke lapisan jaringan. Lapisan ini juga melakukan kolaborasi simpul IoT di jaringan lokal dan jarak pendek.

### c. Network layer

Pada lapisan ini, platform *cloud computing, internet gateway, switching,* dan *routing device*. Beroperasi dengan menggunakan beberapa teknologi terbaru seperti WiFi, LTE, Bluetooth, 3G, Zigbee dll. *Internet gateway* berfungsi sebagai mediator antara berbagai IoT node dengan menggabungkan, memfilter, dan mentransmisikan data ke dan dari sensor yang berbeda.

### d. *Application layer*

Pada lapisan ini untuk menjamin keaslian, tujuan IoT atau penciptaan lingkungan yang cerdas tercapai.




Gambar 1. Arsitektur IoT

### e. Keamanan IoT

Sasaran keamanan Confidentiality, Integrity and Availability (CIA) juga berlaku untuk IoT. Namun, IoT memiliki banyak batasan dan batasan dalam hal komponen dan perangkat, sumber daya komputasi dan daya, dan bahkan sifat IoT yang heterogen serta ada di mana-mana yang menimbulkan kekhawatiran tambahan. Bagian ini terdiri dari dua bagian: *the general security features* yang harus dimiliki IoT, dan *the security issues specific to each layer of the IoT*.

### *f. General security features*

Tantangan keamanan IoT dapat secara luas dibagi menjadi dua kelas yaitu tantangan teknologi dan tantangan Keamanan. Tantangan teknologi muncul karena sifat heterogen dan di mana-mana perangkat IoT, sementara tantangan keamanan terkait dengan prinsip dan fungsi yang harus ditegakkan

untuk mencapai jaringan yang aman. Tantangan teknologi biasanya terkait dengan teknologi nirkabel, skalabilitas, energi, dan sifat terdistribusi, sementara tantangan keamanan membutuhkan kemampuan untuk memastikan keamanan dengan otentikasi, kerahasiaan, keamanan ujung ke ujung, integritas dll. Keamanan harus diberlakukan di IoT selama pengembangan dan siklus hidup operasional semua perangkat dan hub IoT (Mahmoud, dkk. 2015). Ada berbagai mekanisme untuk memastikan keamanan adalah sebagai berikut:

1) Perangkat lunak yang berjalan pada semua perangkat IoT harus diotorisasi. x Ketika perangkat IoT dihidupkan, ia harus terlebih dahulu mengotentikasi dirinya ke dalam jaringan sebelum mengumpulkan atau mengirim data.
2) Karena perangkat IoT memiliki kemampuan komputasi dan memori yang terbatas, firewall diperlukan dalam jaringan IoT untuk memfilter paket yang diarahkan ke perangkat tersebut.
3) Pembaruan dan tambalan pada perangkat harus diinstal dengan cara yang tidak memakan bandwidth tambahan.

**g. *Security Challenges in Each Layer of IoT***

Setiap layer IoT rentan terhadap ancaman dan serangan keamanan. Ini bisa aktif, atau pasif, dan dapat berasal dari sumber eksternal atau jaringan internal . berikut adalah penjelasan tentang masalah keamanan yang berhubungan dengan setiap lapisan IoT:

1) Perception Layer

Ada tiga masalah keamanan di lapisan persepsi IoT. Pertama adalah kekuatan sinyal nirkabel. Sebagian besar sinyal ditransmisikan antara node sensor IoT menggunakan teknologi nirkabel yang keffektifan sinyal dapat terganggu oleh gelombang yang mengganggu. Kedua, simpul sensor pada perangkat IoT dapat dicegat tidak hanya oleh pemilik tetapi juga oleh penyerang karena nodes IoT biasanya beroperasi di lingkungan eksternal , yang mengarah ke serangan fisik pada sensor IoT dan penyerang dapat merusak perangkat keras atau komponen perangkat. Ketiga adalah sifat inheren topologi jaringan yang dinamis karena nodes IoT sering berpindah-pindah tempat yang berbeda. Lapisan persepsi IoT sebagian besar terdiri dari sensor dan RFID, karena kapasitas penyimpanan, konsumsi daya, dan kemampuan komputasi mereka sangat terbatas sehingga rentan terhadap berbagai jenis ancaman dan serangan.

2) *Network layer*

Seperti yang disebutkan sebelumnya, lapisan jaringan IoT juga rentan terhadap serangan DoS. Terlepas dari serangan DoS, musuh juga dapat menyerang kerahasiaan dan privasi di lapisan jaringan dengan analisis traffik, penyadapan, dan pemantauan pasif. Serangan-serangan ini memiliki kemungkinan tinggi terjadi karena mekanisme akses jarak jauh dan pertukaran data perangkat. Lapisan jaringan sangat rentan terhadap serangan Man-in-the-Middle, yang dapat diikuti dengan menyadap. Jika materi kunci perangkat disadap, saluran komunikasi yang aman akan sepenuhnya terganggu. Mekanisme pertukaran kunci dalam IoT harus cukup aman untuk mencegah penyusup menyadap, dan kemudian melakukan pencurian identitas.

3) *Application layer*

Karena IoT masih belum memiliki kebijakan dan standar global yang mengatur interaksi dan pengembangan aplikasi, ada banyak masalah terkait keamanan aplikasi. Aplikasi yang berbeda memiliki mekanisme otentikasi yang berbeda, yang membuat integrasi semuanya sangat sulit untuk memastikan privasi data dan otentikasi identitas. Sejumlah besar perangkat terhubung yang berbagi data akan menyebabkan overhead besar pada aplikasi yang menganalisis data, yang dapat berdampak besar pada ketersediaan layanan. Masalah lain yang harus dipertimbangkan ketika merancang aplikasi di IoT adalah bagaimana pengguna yang berbeda akan berinteraksi dengan mereka, jumlah data yang akan diungkapkan, dan siapa yang akan bertanggung jawab untuk mengelola aplikasi ini.memiliki alat untuk mengontrol data apa yang ingin mereka sampaikan dan harus mengetahui bagaimana data akan digunakan, oleh siapa dan kapan.

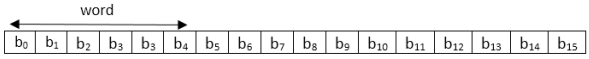
h. ***Advanced Encryption Standard* (AES)**

AES (*Advanced Encryption Standard*) adalah blok chipertext simetrik yang dapat mengenkripsi (encipher) dan dekripsi (decipher) infoermasi. Enkripsi merubah data

yang tidak dapat lagi dibaca disebut ciphertext; sebaliknya dekripsi adalah merubah ciphertext data menjadi bentuk semula yang kita kenal sebagai plaintext. Algoritma AES is mengunkan kunci kriptografi 128, 192, dan 256 bits untuk mengenkrip dan dekrip data pada blok 128 bits. AES memiliki ukuran block yang tetap sepanjang 128 bit dan ukuran kunci sepanjang 128, 192, atau 256 bit. Berdasarkan ukuran block yang tetap, AES bekerja pada matriks berukuran 4x4 di mana tiap-tiap sel matriks terdiri atas 1 byte (8 bit)(Aminatus, Munadi, and Bisono 2018).

Tabel 1. Unit data AES

Bbyte = [b0 b1 b2 b3 b4 b5 b6 b7 b8]




$S_{0.0}=b_0$ $S_{0.1}=b_4$ $S_{0.2}=b_8$ $S_{0.3}=b_{12}$
$S_{1.0}=b$ $S_{1.1}=b$ $S_{1.2}=b$ $S_{1.3}=b$
$S_{2.0}=b$ $S_{2.1}=b$ $S_{2.2}=b$ $S_{1.4}=b$
$S_{3.0}=b$ $S_{3.1}=b$ $S_{3.2}=b$ $S_{1.5}=b$

Secara garis besar proses enkripsi AES terdiri dari 4 jenis transformasi, yaitu SubBytes, ShiftRows, MixColumns dan AddRoundKey, sedangkan pada ronde terahir tidak dilakukan transformasi Mixcolumns. Skema enkripsi AES dapat dilihat pada Gambar 2. Proses dekripsi AES menggunakan transformasi invers yaitu, InvSubBytes, InvShiftiRows, InvsMixColumns. AddRoundKey merupakan transformasi yang bersifat self-invers dengan syarat menggunakan kunci yang sama(Darwis, dkk. 2018).

Tabel 4. Panjang kunci dan ronde AES

| Panjang Kunci AES (bit) | Jumlah Ronde (Nr) |
|---|---|
| 128 | 10 |
| 192 | 12 |
| 256 | 14 |

## 3. HASIL DAN PEMBAHASAN

Berdasarkan penelitian dari (Darwis, dkk. 2018) berdujudul "Kombinasi Gifshuffle, Enkripsi AES dan Kompresi Data Huffman Untuk Meningkatkan Keamanan Data" berhasil menggabungkan metode kriptografi AES dengan metode steganografi gifsuffle. Hasil pengujian imperceptibility menunjukkan 85% responden tidak dapat membedakan gambar asli dengan gambar yang telah disisipi pesan. Pemisahan gambar dengan pesan dapat dilakukan dengan akurasi 100% dan proses dekripsi pesan cipher-text menjadi plain-text juga dapat dilakukan dengan sempurna.

Tabel 3. Hasil uji Enkripsi dan Dekripsi

| No. | *Plain text* (Hexa) | Enkripsi (*seconds*) | Dekripsi (*seconds*) |
|---|---|---|---|
| 1 | 4e 55 52 55 4C 48 4F 54 49 4D 41 48 55 4C 55 4C | 0.322144 | 0.210373 |
| 2 | 4B 41 4B 4B 44 45 44 49 62 61 69 6B 48 41 54 49 | 0.132969 | 0.124793 |
| 3 | 50 41 4B 4E 47 41 44 69 52 61 4E 42 50 4B 4B 55 | 0.120812 | 0.153897 |
| 4 | 6E 75 72 75 6C 4C 41 47 49 62 65 6C 61 6A 61 72 | 0.123331 | 0.139135 |
| 5 | 6D 41 54 45 6D 61 74 69 6B 41 41 53 49 4B 6C 6F | 0.115934 | 0.135554 |
| 6 | 41 6c 6c 48 6B 75 4D 41 48 41 62 65 73 40 52 | 0.150456 | 0.160964 |
| 7 | 4D 41 4D 41 4B 42 41 50 41 4B 53 45 68 61 74 7A | 0.120771 | 0.144201 |
| 8 | 6D 61 73 48 34 31 32 69 4D 33 34 4B 79 75 6E 7c | 0.117455 | 0.146964 |

Menurut penelitian dari (Arief Agung Gumelar, dkk. 2017)yang berjudul "Sistem Keamanan ATM dengan Mengguanakan Enkripsi AES Pada Kartu ATM". Menjelaskan bahwa Dengan menggunakan aplikasi cryptool disini kita dapat melihat cara kerja enkripsi algoritma AES 128 bit bekerja dan dengan aplikasi ini dapat mengubah bentuk *plain text* menjadi *chiper text* serta metode yang di gunakan aplikasi ini adalah metode AES dan bagaimana enkripsi AES bekerja.

### a. Struktur Enkripsi Pada AES

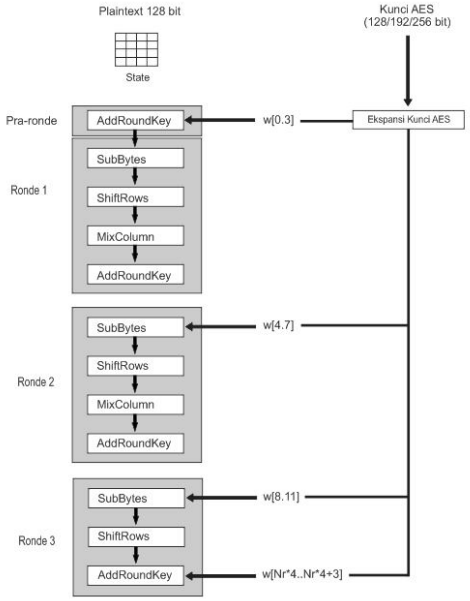
Dalam enkripsi struktur AES merupakan transformasi terhadap State sebuah teks asli berupa plaintext dalam blok 128 bit terlebih dahulu akan diorganisir sebagai State. enkripsi AES adalah suatu transformasi terhadap State yang dilakukan secara berulang dalam beberapa ronde state yang menjadi keluaran ronde k menjadi masukan untuk ronde ke-k+1.

Secara garis besar desain enkripsi aes diberikan oleh gambar 2. mulanya sebuah plaintext di organisasi sebagai sebuah state

setelah itu sebelum ronde 1 mulai teks asli yang berupa plaintext dicampur dengan kunci ronde ke-0 pada bagian ini disebut sebagai *AddRoundKey* kemudian ronde ke 1 sampai dengan ronde ke (NR-1) dengan Nr merupakan jumlah ronde menggunakan 4 jenis transformasi, yaitu *SubBytes, ShiftRows, MixColumns dan AddRoundKey*. Pada ronde ke-Nr akan dilakukan transformasi serupa dengan ronde lain namun tanpa MixColumns.

**b. Struktur Deskripsi Pada AES**

Dalam proses enkripsi yang dilakukan sebelumnya untuk mengubah sebuah text asli (*plaintext*) menjadi *ciphertext* didasarkan pada beberapa langkah-langkah yang disebut sebagai ronde, namun jika dalam proses deskripsi ada sedikit perbedaan karena pada dasarnya proses ini adalah melakukan dekrip dari sebuah kode enkripsi yang di sebut ciphertext menjadi text aslinya lagi (*plaintext*). Algoritma dekripsi AES menggunakan transformasi invers semua transformasi dasar yang digunakan pada algoritma enkripsi AES. Sehingga dalam transformasi dasar AES memiliki invers, yaitu *InvSubBytes, InvShiftRows dan InvMixColumns*.




Gambar 2. Struktur Enkripsi AES

*AddRoundKey* adalah transformasi yang sifatnya *self-invers* memiliki syarat menggunakan *Key* yang sama seperti ditunjukkan pada gambar 2. diatas adalah algoritma deskripsi untuk AES.

**Algoritma** Enkripsi AES
**Input:** P,K {Teks Asli 16 bytes, kunci AES(128,192,256 bit)}
**Output:** CT {Teks sandi 16 bytes}
(*Nr*,*w*) ← **EkspansiKunci**(K) {*Nr* : Jumlah Ronde, *w* : larik bytes kunci ronde}
CT = P
**AddRoundKey**(CT,*w*[0..3])
For *i* = 1 → *Nr* **do**
**SubBytes**(CT)
**ShiftRows**(CT)
**if** ≠ *Nr* **then**
**MixColumns**(CT)
**end if**

**AddRoundKey**(CT,*w*[(*i**4)..(*i**4)+3])
**end for**

**Algoritma** Deskripsi AES
**Input:** CT,K{Teks sandi 16 bytes, kunci AES (128,192, 256 bit)}
**Output:** P {Teks asli 16 bytes}
(*Nr*,w) ← **EkspansiKunci**(K) {*Nr* : Jumlah Ronde, *w* : larik bytes kunci ronde}
P=CT
**AddRoundKey**(P,w[*Nr**4..*Nr**4-3])
For i = 1 → *Nr* **do**
**InvSubBytes**(P)
**InvShiftRows**(P)
**AddRoundKey**(P,w[(*Nr*-*i*) *4)..((*Nr*-*i*)*4+3])
**if** ≠ *Nr* **then**

**MixColumns**(P)
**end if**
**end fo**

- **Transformasi-transformasi AES**

Algoritma enkripsi AES menggunakan 4 jenis transformasi: substitusi yang disebut dengan **SubBytes**, permutasi yang disebut dengan **ShiftRows**, pencampuran yang disebut dengan **MixColumns**, dan penambahan kunci yang disebut **AddRoundKey.**

4) SubBytes

AES menggunakan substitusi nonlinier pada ukuran **byte** yang disebut **SubBytes.** Setiap elemen pada *state* dari elemn $s_{(0,0)}$ sampai dengan $s_{(3,3)}$ dikenakan transformasi **SubBytes**.

Transformasi **SubBytes** dapat menggunakan tabel substitusi, yaitu dengan cara menginterpretasikan *byte* masukan $s_{i,j}$ sebagai 2 bilangan heksadesimal, kemudian digit kiri menunjukkan indeks baris dan digit kanan menunjukkan indeks kolom di tabel substitusi. Nilai *byte* pada tabel substitusi yang dirujuk oleh indeks baris dan kolom menjadi nilai yang mensubstitusi $s_{i,j}$. Tabel substitusi untuk **SubBytes** diberikan oleh

Tabel 5. sedangkan tabel invers substitusi **SubBytes** (transfor-masinya diberi nama **InvSubBytes**) diberikan oleh tabel 6.

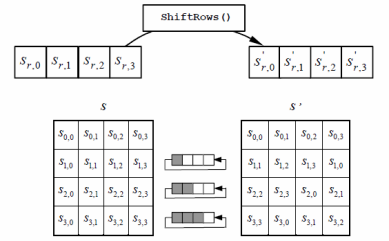
Tabel 5. Tabel substitusi untuk transformasi *SubBytes*

| u \ v | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 63 | 7c | 77 | 7b | f2 | 6b | 6f | c5 | 30 | 01 | 67 | 2b | fe | d7 | ab | 76 |
| 1 | ca | 82 | c9 | 7d | fa | 59 | 47 | f0 | ad | d4 | a2 | af | 9c | a4 | 72 | c0 |
| 2 | b7 | fd | 93 | 26 | 36 | 3f | f7 | cc | 34 | a5 | e5 | f1 | 71 | d8 | 31 | 15 |
| 3 | 04 | c7 | 23 | c3 | 18 | 96 | 05 | 9a | 07 | 12 | 80 | e2 | eb | 27 | b2 | 75 |
| 4 | 09 | 83 | 2c | 1a | 1b | 6e | 5a | a0 | 52 | 3b | d6 | b3 | 29 | e3 | 2f | 84 |
| 5 | 53 | d1 | 00 | ed | 20 | fc | b1 | 5b | 6a | cb | be | 39 | 4a | 4c | 58 | cf |
| 6 | d0 | ef | aa | fb | 43 | 4d | 33 | 85 | 45 | f9 | 02 | 7f | 50 | 3c | 9f | a8 |
| 7 | 51 | a3 | 40 | 8f | 92 | 9d | 38 | f5 | bc | b6 | da | 21 | 10 | ff | f3 | d2 |
| 8 | cd | 0c | 13 | ec | 5f | 97 | 44 | 17 | c4 | a7 | 7e | 3d | 64 | 5d | 19 | 73 |
| 9 | 60 | 81 | 4f | dc | 22 | 2a | 90 | 88 | 46 | ee | b8 | 14 | de | 5e | 0b | db |
| 10 | e0 | 32 | 3a | 0a | 49 | 06 | 24 | 5c | c2 | d3 | ac | 62 | 91 | 95 | e4 | 79 |
| 11 | e7 | c8 | 37 | 6d | 8d | d5 | 4e | a9 | 6c | 56 | f4 | ea | 65 | 7a | ae | 08 |
| 12 | ba | 78 | 25 | 2e | 1c | a6 | b4 | c6 | e8 | dd | 74 | 1f | 4b | bd | 8b | 8a |
| 13 | 70 | 3e | b5 | 66 | 48 | 03 | f6 | 0e | 61 | 35 | 57 | b9 | 86 | c1 | 1d | 9e |
| 14 | e1 | f8 | 98 | 11 | 69 | d9 | 8e | 94 | 9b | 1e | 87 | e9 | ce | 55 | 28 | df |
| 15 | 8c | a1 | 89 | 0d | bf | e6 | 42 | 68 | 41 | 99 | 2d | 0f | b0 | 54 | bb | 16 |

| u \ v | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 52 | 09 | 6a | d5 | 30 | 36 | a5 | 38 | bf | 40 | a3 | 9e | 81 | f3 | d7 | fb |
| 1 | 7c | e3 | 39 | 82 | 9b | 2f | ff | 87 | 34 | 8e | 43 | 44 | c4 | de | e9 | cb |
| 2 | 54 | 7b | 94 | 32 | a6 | c2 | 23 | 3d | ee | 4c | 95 | 0b | 42 | fa | c3 | 4e |
| 3 | 08 | 2e | a1 | 66 | 28 | d9 | 24 | b2 | 76 | 5b | a2 | 49 | 6d | 8b | d1 | 25 |
| 4 | 72 | f8 | f6 | 64 | 86 | 68 | 98 | 16 | d4 | a4 | 5c | cc | 5d | 65 | b6 | 92 |
| 5 | 6c | 70 | 48 | 50 | fd | ed | b9 | da | 5e | 15 | 46 | 57 | a7 | 8d | 9d | 84 |
| 6 | 90 | d8 | ab | 00 | 8c | bc | d3 | 0a | f7 | e4 | 58 | 05 | b8 | b3 | 45 | 06 |
| 7 | d0 | 2c | 1e | 8f | ca | 3f | 0f | 02 | c1 | af | bd | 03 | 01 | 13 | 8a | 6b |
| 8 | 3a | 91 | 11 | 41 | 4f | 67 | dc | ea | 97 | f2 | cf | ce | f0 | b4 | e6 | 73 |
| 9 | 96 | ac | 74 | 22 | e7 | ad | 35 | 85 | e2 | f9 | 37 | e8 | 1c | 75 | df | 6e |
| 10 | 47 | f1 | 1a | 71 | 1d | 29 | c5 | 89 | 6f | b7 | 62 | 0e | aa | 18 | be | 1b |
| 11 | fc | 56 | 3e | 4b | c6 | d2 | 79 | 20 | 9a | db | c0 | fe | 78 | cd | 5a | f4 |
| 12 | 1f | dd | a8 | 33 | 88 | 07 | c7 | 31 | b1 | 12 | 10 | 59 | 27 | 80 | ec | 5f |
| 13 | 60 | 51 | 7f | a9 | 19 | b5 | 4a | 0d | 2d | e5 | 7a | 9f | 93 | c9 | 9c | ef |
| 14 | a0 | e0 | 3b | 4d | ae | 2a | f5 | b0 | c8 | eb | bb | 3c | 83 | 53 | 99 | 61 |
| 15 | 17 | 2b | 04 | 7e | ba | 77 | d6 | 26 | e1 | 69 | 14 | 63 | 55 | 21 | 0c | 7d |

5) ShiftRows dan InvshiftRows

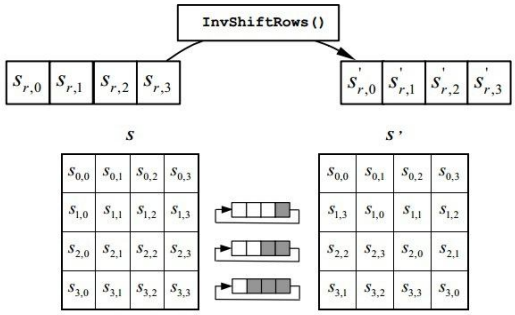
Selain menggunakan substitusi untuk mengganti nilai pada elemen *state*, AES menggunakan permutasi pada *state*. Transformasi permutasi pada *state* disebut dengan transformasi **ShiftRows. ShiftRows** dilakukan dengan menjalan operasi *circular shift left* sebanyak i pada baris ke-i pada *state*. Ilustrasi transformasi **ShiftRows** diberikan oleh Gambar 2.




Gambar 2**.** Transformasi *ShiftRows*

Transformasi **ShiftRows** merupakan jenis transformasi permutasi, yaitu pengubahan posisi elemen pada *state* tanpa mengubah nilainya. Transformasi **ShiftRows** terlihat sederhana jika dilihat melalui representasi *state*. Namun, karena *state* adalah representasi blok dengan oriental per kolom menjadikan transformasi **ShifRows** menjadi rumit jika dilihat dari sudut pandang blok.

Tranformasi invers terhadap **ShiftRows** disebut **InvShiftRows.** Transformasi **InvShiftRows** terhadap sebuah *state* menggunakan operasi (*circular shift right*) pada tiap barisnya yang banyak gesernya sesuai dengan indeks baris seperti yang ditunjukkan oleh Gambar 3.




Gambar 3**.** Transformasi *InvShiftRows*

6) MixColumns

Tujuan transformasi **MixColumn** adalah mencampur nilai kolom pada *state* pada satu elemen *state* keluaran. Untuk melakukan pencampuran itu, transformasi **MixColumn** menggunakan operasi perkalian matriks dengan operasi perkalian matriksdengan operasi perkalian.

$$\begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

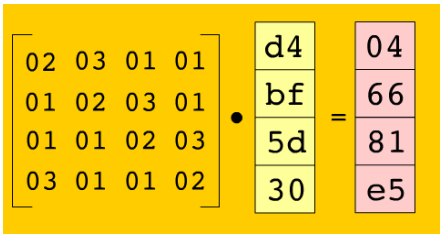
Gambar 4. Formula penyelesaian *MixColumn*

| | | | |
|---|---|---|---|
| d4 | e0 | b8 | 1e |
| bf | b4 | 41 | 27 |
| 5d | 52 | 11 | 98 |
| 30 | ae | f1 | e5 |

| | | | |
|---|---|---|---|
| 02 | 03 | 01 | 01 |
| 01 | 02 | 03 | 01 |
| 01 | 01 | 02 | 03 |
| 03 | 01 | 01 | 02 |

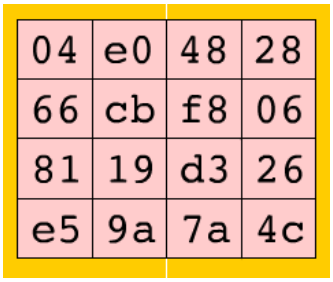
Gambar 5. Contoh *MixColumn* dan nilai matriks

Gambar matriks diatas dikalikan berurutan dimulai dengan mengambil pada kolom ke-1 ($a_{(0,0)}$ sampai $a_{(0,3)}$) dengan matriks 1 dilanjutkan hingga kolom terakhir ($a_{(3,0)}$ sampai $a_{(3,3)}$) dengan matriks 4.




Gambar 6. *Metode perkalian*

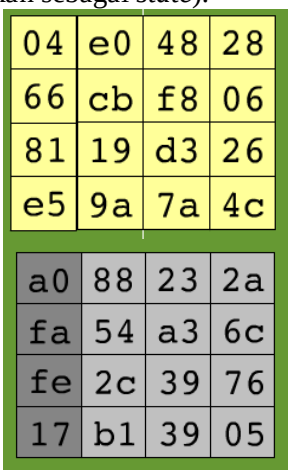
Dan akan menemukan hasil untuk melanjutkan ke metode selanjutnya yang ada pada Gambar 7.




Gambar 7. *Hasil dari perkalian*

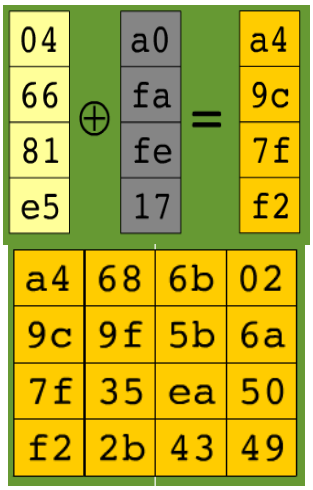
7) AddRoundKey

Transformasi keempat yang digunakan pada penyandian AES adalah transformasi **AddRoundKey**. Transformasi **AddRoundKey** mencampur sebuah OR(⊕). Setiap elemen pada *state* masukan yang merupakan sebuah byte dikenakan operasi eksklusif OR dengan bytepada posisi yang sama di kunci ronde (kunci ronde direpresentasikan sebagai *state*).




Gambar 8. *Hasil MixColumn dan RoundKey*

Kalikan kolom pertama *Round Key* dengan kolom pertama hasil *MixColumn* begitu juga setelahnya




Gambar 9. *Meode XOR*

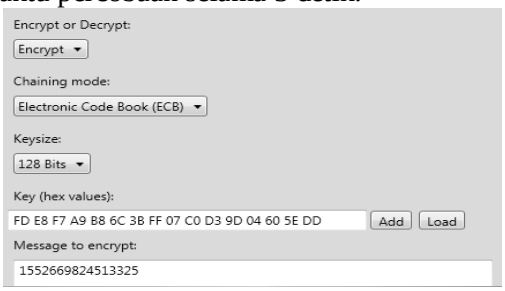
Kalikan begitu juga seterusnya hingga menemukan hasil dari round pertama yang ada dan akan ketemu hasil akhir setelah melakukan *looping* hingga 9 kali.

- **Proses Enkripsi**

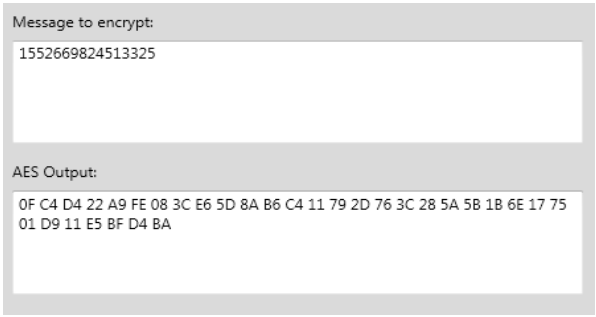
Dalam proses enkripsi di dalam tulisan (Arief Agung Gumelar, dkk. 2017) menggunakan sebuah program yang bernama cryptotool.


Gambar 10. Program Cryptool

Beberapa parameter di tentukan sebelum melakukan enkripsi seperti *KeySize, Key dan Chaining Mode* kemudian proses enkripsi dilakukan dengan menghabiskan waktu percobaan selama 3 detik.




Gambar 11. Proses Enkripsi Data




Gambar 12. Hasil Enkripsi Data

## 4. KESIMPULAN DAN SARAN

### a. Kesimpulan

Begitu banyaknya tantangan terutama dalam bidang kemanan IoT yang masih bisa dijadikan bahan penelitian dan enkripsi AES adalah salah satu solusi kemanan IoT untuk mengekrepsi data-data yang tidak boleh diakses oleh sembarang orang terutama data pribadi masing-masing pengguna IoT. Dengan enkripsi AES yang memiliki empat kali proses enkripsi keamanan *chipper text* yang dihasilkan oleh enkripsi AES sudah cukup untuk mengenkripsi data pada IoT.

### b. Saran

Untuk penulisan paper selanjutnya lebih menjabarkan Algoritma apa saja yang dapat melakukan enkripsi untuk memberikan keamanan dalam IoT.

## 5. REFERENSI

Aminatus, Jorjiana, Ir Rendy Munadi, and Gustommy Bisono. 2018. "IMPLEMENTASI DAN ANALISA SISTEM KEAMANAN DI JARINGAN SENSOR IMPLEMENTATION AND ANALYSIS OF SECURITY SYSTEM IN WIRELESS SENSOR." 5(1): 546–54.

Arief Agung Gumelar, Latief Adam Busyairi, Muhammad Fajrian Noor. 2017. "SISTEM KEAMANAN ATM DENGAN MENGGUNAKAN." : 1–6.

Darwis, Dedi et al. 2018. "KOMBINASI GIFSHUFFLE , ENKRIPSI AES DAN KOMPRESI DATA HUFFMAN COMBINATION OF GIFSHUFFLE , AES ENCRYPTION AND HUFFMAN Proses Pengamanan Data Dapat Dilihat Pada." 5(4): 389–94.

Mahmoud, Rwan, Tasneem Yousuf, Fadi Aloul, and Imran Zualkernan. 2015. "Internet of Things ( IoT ) Security : Current Status , Challenges and Prospective Measures." : 336–41.

Meutia, Ernita Dewi. 2015. "Internet of Things – Keamanan Dan Privasi."