Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X
Vol. 2, No. 12, Desember 2018, hlm. 6729-6736 http://j-ptiik.ub.ac.id

Implementasi Algoritme *Advance Encryption Standard* (AES) pada Enkripsi dan Dekripsi QR-Code

Dwi Qunita Putri Ambeq Paramarta[1], Ari Kusyanti[2], Mahendra Data[3]

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]dwiqunitaputri@gmail.com, [2]ari.kusyanti@ub.ac.id, [3]mahendra.data@ub.ac.id

Abstrak

Keamanan data merupakan masalah yang sangat penting dalam perkembangan teknologi saat ini. Oleh sebab itu dibutuhkan sebuah cara yang dapat menjaga keamanan merujuk pada perlindungan informasi dari penyingkapan pihak yang tidak sah. Salah satu mekanisme untuk meningkatkan keamanan data adalah dengan menggunakan teknik kriptografi. Ada berbagai macam algoritme dalam kriptografi salah satunya adalah *Algoritme Advance Encryption Standard.* Skripsi ini menggunakan *Algoritme AES* dengam ukuran ekspansi key 128 bit yang akan beroperasi dalam sebuah *array* 4x4. Pada proses *state* enkripsi akan melalui beberapa tahapan yakni *Addroaundkey*, *Subbyte*, *Shiftrows*, dan *Mixcolumns* sebanyak 10 kali putaran. Namun pada putaran terakhir tidak dilakukan lagi proses *Mixcolumns* langsung ke proses *Addraoundkey*, dan untuk proses dekripsi merupakan proses kebalikan dari proses enkripsi yakni *InvAddrows*, *InvShiftrows*, *InvSubbyte*, *dan InvMixcolumns* menggunakan kunci *round* yang sama dengan proses enkripsi. AES diimplementasikan dalam bahasa pemrograman PHP dan diterapkan pada QR *Code* karena merupakan sebuah teknologi *labelling* yang dapat menyimpan data dalam bentuk pola yang dapat diisi dengan informasi. Dari hasil implementasi algoritme AES dapat disimpulkan bahwa aplikasi ini dapat mengekripsi semua jenis karakter berupa *string*, huruf, angka, dan simbol. Pada saat mendekripsi QR *Code* aplikasi akan mengaktifkan fungsi kamera dan melakukan *scanning* QR *Code* yang akan menjadi *plaintext* kembali. Waktu eksekusi enkripsi dan dekripsi AES adalah 0.0034 detik untuk proses enkripsi dan untuk proses dekripsi membutuhkan waktu 0.0029 detik.

Kata kunci: *AES*, Enkripsi, Dekripsi, dan QR-Code.

Abstract

Data security are very important in today's technological development. Therefore, it is necessary to find a way to protect the confidentiality and the security from unauthorized accesses. One of the mechanism to increase data security is to use cryptography. There are many form of cryptography, one of them is Advance Encryption Standard Algorithm. This thesis uses AES Algorithm with the size of 128 bit expansion key. That will operate in a 4x4 array. In the state encryption process will go through several stages of Addroaundkey, Subbyte, Shiftrows, and Mixcolumns 10 times round. But in the last round Mixcolumns no longer process directly into the Addraoundkey process, and for the decryption process is a reverse process of the encryption process that InvAddrows, InvShiftrows, InvSubbyte, and InvMixcolumns use the same round key with the encryption process. AES is implemented in PHP programming language and applied to QR-Code because it is a labeling technology that can store data in the form of patterns that can be filled with information. From the results of the implementation of AES Algorithm can be concluded that this application can encrypt all types of characters in the form of strings, alphabet, numbers, and symbols. When decrypting QR-Code the application will activate the camera function and perform QR-Code scanning that will be plaintext again. The execution time of AES encryption and decryption is 0.0034 seconds for the encryption process and for the decryption process takes 0.0029 seconds.

Keywords: *AES, Encryption, Decryption, and QR-Code.*

1. PENDAHULUAN

Keamanan data merupakan masalah yang sangat penting dalam perkembangan teknologi

saat ini. Oleh sebab itu dibutuhkan sebuah cara yang dapat menjaga kerahasiaan dan keamanan merujuk pada perlindungan informasi dari penyingkapan pihak yang tidak sah. Salah satu mekanisme untuk meningkatkan keamanan data adalah dengan menggunakan teknik kriptografi. Ada berbagai macam algoritme dalam kriptografi salah satunya adalah *Algoritme Advance Encryption Standard,* data-data yang disimpan diubah sedemikian rupa sehingga tidak mudah dibaca. Enkripsi adalah proses yang dilakukan untuk merubah suatu informasi sehingga tidak dapat dibaca oleh orang yang tidak bertanggung jawab. Sebaliknya, proses dekripsi merupakan suatu proses yang mengembalikan informasi yang sudah dienkripsi menjadi bisa dibaca kembali. (Daemen & Rjmen, 03 September 1999)

Metode di atas bisa diterapkan pada QR-*Code*, QR-Code merupakan sebuah teknologi labelling yang dapat menyimpan data dalam bentuk pola yang dapat diisi dengan informasi. QR-Code merupakan bentuk evolusi dari kode batang dari satu dimensi menjadi dua dimensi yang dikembangkan oleh Denso Wave. Pengenalan pola dilakukan dengan mendeteksi marker atau tanda yang telah diisi dengan informasi yang dibutuhkan. QR merupakan singkatan dari Quick Response. Tujuannya adalah untuk menyampaikan informasi dengan cepat dan mendapatkan respon yang cepat pula. Berbeda dengan kode batang yang hanya menyimpan informasi secara horizontal, QR-code mampu menyimpan informasi secara horizontal dan vertikal. Oleh karena itu, QR Code dapat menampung informasi yang lebih banyak misalnya dalam bentuk URL, teks, angka, dll. Karenanya, dengan menggunakan QR Code, kita dapat menyimpan informasi mengenai nomor surat, klasifikasi dokumen, pengirim dan perihal (Denso, 2013). Kelebihan QR Code dibandingkan dengan Barcode adalah: (1) kapasitas atau panjang kata lebih banyak; (2) tipe data yang disimpan pada QR Code beragam dapat berupa angka atau huruf atau gabungan keduanya; (3) QR Code dapat dibaca dari segala arah sehingga kemungkinan gagal dalam membaca sangat kecil; (4) memiliki ketahanan hingga 30%. Sehingga apabila QR Code mengalami kerusakan hingga 30 % dapat tetap terbaca (Denso, 2013). Dengan kelebihan dan manfaat yang dimiliki, QR Code dapat digunakan sebagai sarana identifikasi surat pada saat surat masuk di bagian tata usaha suatu instansi. Otentikasi aman, dicapai dengan menggunakan algoritme penyembunyian data dengan QR *code.*

Pada penelitian yang dilakukan oleh (Sholeh & Muharom, 2016) yang berjudul "Smart Presensi Menggunakan QR-*Code* Dengan Enkripsi Vigenere" menggunakan algoritme kriptografi klasik yang ditemukan oleh Giovan Battista Bellaso yaitu algoritme *Vigenere* untuk perlindungan data atau informasi menggunakan QR-*Code*. Pada penelitian yang dilakukan oleh (Harahap, 2016) yang berjudul "Analisis Perbandingan Algoritma Kriptografi Klasik Vigenere Cipher dan One Time Pad" menyimpulkan bahwa dari segi keamana algoritme *Vigenere* ini memiliki ketahanan yang kurang kuat dari segi kunci. Karena algoritma *Vigenere* menggunakan kunci yang selau berulang sepanjang pesan yang akan dienkripsi. Sehingga memungkinkan terjadinya kecoboran informasi. Pada penelitian yang dilakukan oleh (Sumandri, 2017) yang berjudul "Studi Model Algoritma Kriptografi Klasik dan Modern" yang menyimpulkan bahwa terdapat 3 model kriptografi klasik yaitu Caesar Cipher, Vigenere Chiper dan Hill Cheper. Pada penelitian tersebut juga dijelaskan kelemahan dari Vigenere Cipher adalah penggunaan kunci yang berulang-ulang. Sementara kriptografi modern menggunakan mode bit biner (0 dan 1) yang di bentuk dari kode ASCII, sehingga mempunyai tingkat kesulitan yang kompleks. Kekuatan kriptografi modern ada pada kuncinya (*key*). Kunci yang digunakan pada kriptografi modern terdiri dari tiga jenis yakni: simetris; asimetris; dan hibrida. Contoh kriptografi modern yaitu MD5, RC4, AES dan lain-lain.

Dari hasil penelitian diatas dapat ditarik kesimpulan bahwa algoritme *vigenere* memiliki ketahanan yang kurang kuat dalam pembuatan kunci. Metode yang digunakan algoritme *vigenere* dalam pembuatan kunci yaitu diambil dari substitusi polyalphabetic di mana setiap alfabet bisa diganti dengan beberapa huruf cipher.

AES memiliki tingkat keamanan yang lebih baik dibandingkan dengan algoritme yang ada sebelumnya baik dari segi kunci maupun ukuran blok, jadi AES memiliki ketahanan yang lebih baik dalam mengamankan data jika dibandingkan dengan algoritme viginere. Berdasarkan alasan yang telah dipaparkan di atas, pada penelitian ini membahas tentang implementasi algoritme *Advanced Encryption Standard* (AES) 128 bit dengan *key* dan *plaintext* 128 bit untuk enkripsi dan dekripsi pada QR

Code.

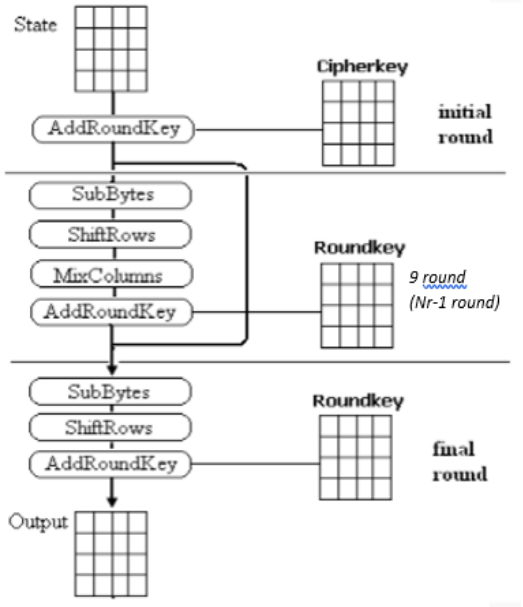
2. DASAR TEORI

2.1 Algoritme AES

Advanced Encryption Standard (AES) merupakan teknik atau algoritme kriptografi penyandian pesan yang menggunakan teknik blok simetris. Algoritme ini dikembangkan oleh dua kriptografer yang berasal dari Belgia, yaitu Dr. Joan Daemen dan Dr. Vincent Rijmen pada tahun 1997. Mereka berdua mengajukan algoritme ini sebagai kandidat AES, dan pada November 2001 disahkan sebagai proposal terpilih bagi AES oleh National Institute of Standard and Technology (NIST) (Daemen & Rjmen, 03 September 1999).

AES sendiri memiliki tipe yang terbagi berdasarkan panjang blok data seperti AES-128, AES-192, AES-256 dimana masing-masing AES memiliki panjang blok sebanyak 128 bit, 192 bit, dan 256 bit. Berikut ilustrasi enkripsi dan dekripsi AES:

A. Enkripsi AES

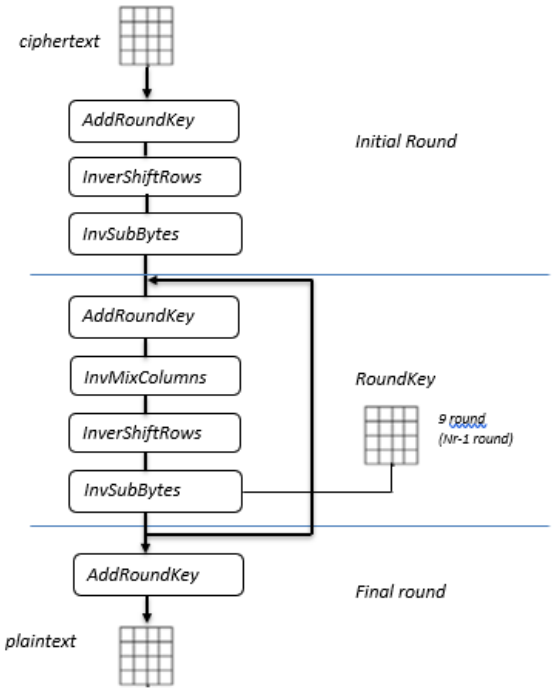


Gambar 2.1 Enkripsi AES

Sumber: Munir (2006)

Gambar 2.1 menjelaskan tentang alur dari proses enkripsi AES 128 bit dengan *key* dan *plaintext* 128 bit yaitu pertama *Addroaundkey*, *Subbyte*, *Shiftrows*, dan *Mixcolumns* sebanyak 10 kali putaran. Namun pada putaran terakhir tidak dilakukan lagi proses *Mixcolumns* langsung ke proses *Addraoundkey*.

B. Dekripsi AES



Gambar 2.2 Dekripsi AES

Sumber: Munir (2006)

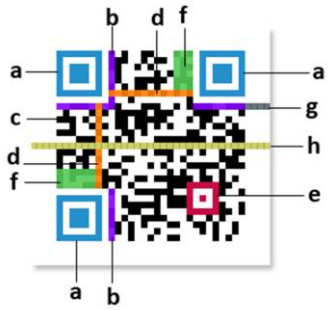
Gambar 2.2 menjelaskan tentang alur dari proses dekripsi AES proses ini merupakan proses kebalikan dari proses enkripsi yakni *InvAddrows*, *invShiftrows*, *InvSubbyte*, *dan InvMixcolumns*, dengan kunci *round* yang sama dengan proses enkripsi.

2.2 QR-Code

Quick Response Code sering di sebut *QR Code* atau Kode *QR* adalah semacam simbol dua dimensi yang dikembangkan oleh Denso Wave yang merupakan anak perusahaan dari Toyota sebuah perusahaan Jepang pada tahun 1994. Tujuan dari *QR Code* ini adalah untuk menyampaikan informasi secara cepat dan juga mendapat tanggapan secara cepat. *QR Code* adalah perkembangan dari *barcode* atau kode batang

yang hanya mampu menyimpan informasi secara horizontal sedangkan *QR Code* mampu menyimpan informasi lebih banyak, baik secara horizontal maupun vertikal.

A. Anatomi QR-Code



Gambar 2.3 Anatomi QR-Code

Sumber: Ariadi (2011)

Beberapa penjelasan anatomi *QR Code* Menurut (Ariadi, 2011) antara lain :

a. *Finder Pattern* berfungsi untuk identifikasi letak *QR Code*.

b. *Format Information* berfungsi untuk informasi tentang *error correction level* dan *mask pattern*.

c. *Data* berfungsi untuk menyimpan data yang dikodekan.

d. *Timing Pattern* merupakan pola yang berfungsi untuk identifikasi koordinat pusat *QR Code*, berbentuk modul hitam putih.

e. *Alignment Pattern* merupakan pola yang berfungsi memperbaiki penyimpangan *QR Code* terutama distorsi non linier.]

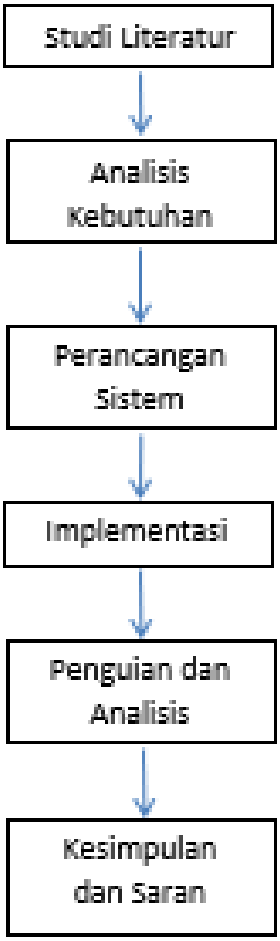
f. *Version Information* adalah versi dari sebuah *QR Code*.

g. *Quiet Zone* merupakan daerah kosong di bagian terluar *QR Code* yang mempermudah mengenali pengenalan *QR* oleh sensor *CCD*.

h. *QR Code version* adalah versi dari *QR Code* yang digunakan.

3. METODOLOGI

Bab ini menjelaskan langkah-langkah yang digunakan dalam penelitian, yaitu studi literatur, analisis kebutuhan, perancangan sistem, implementasi, pengujian dan analisis, kesimpulan dan saran. Gambar 2.4 menjelaskan tentang runtutan pengerjaan penelitian.

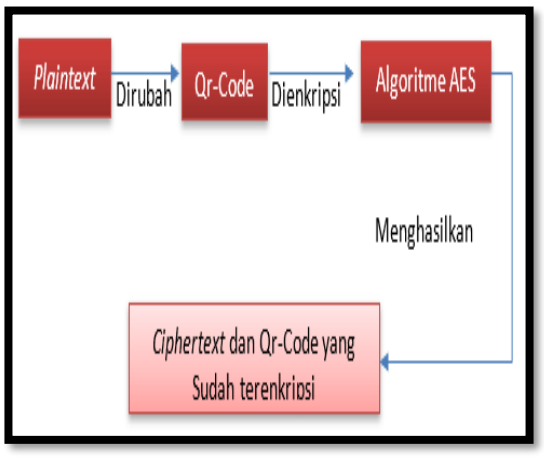


Gambar 2.4 Diagram alir penelitian

4. PERANCANGAN

4.1 Perancangan Umum Sistem

Perancangan umum sistem merupakan tahap awal dari perancangan perangkat lunak.

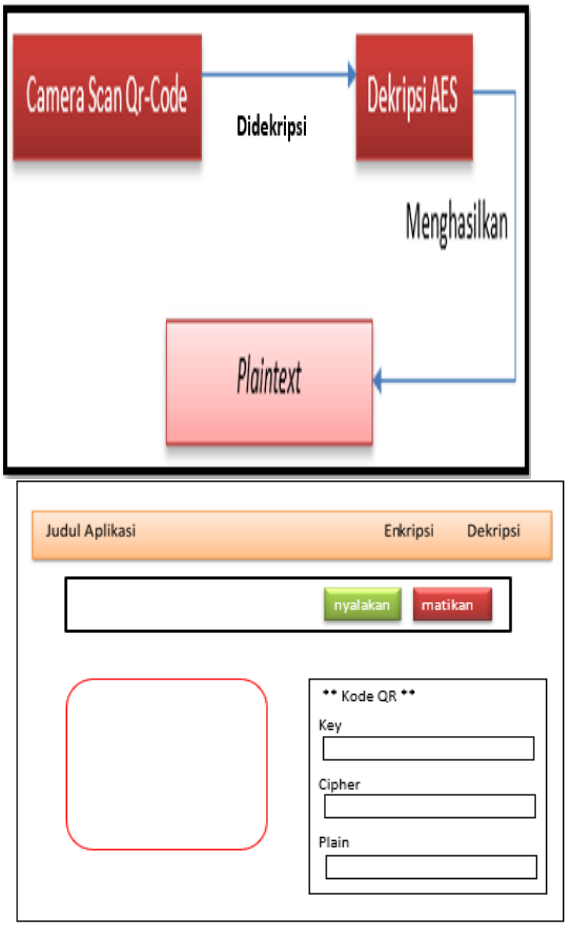


Perancangan dilakukan dengan merepresentasikan arsitektur sistem secara umum. Gambar 4.1 merupakan rancangan enkripsi yang berfungsi untuk merubah dari *plaintext* menjadi QR-*Code* dan dienkripsi AES setelah itu menghasilkan *ciphertext* dan QR-*Code* yang telah dienkripsi. Selanjutnya Gambar 4.2 berfungsi sebagai dekripsi yang bermula dari gambar QR *Code* discan lalu didekripsi menggunakan algoritme AES, dekripsi ini berfungsi untuk membalikkan *ciphertext* menjadi *plaintext*.

Gambar 4.1 Perancangan enkripsi

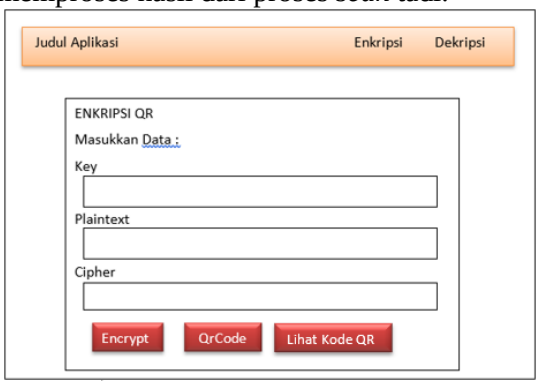
Gambar 4.2 Perancangan dekripsi

4.2 Perancangan Antarmuka



Gambar 4.3 Perancangan antarmuka dekripsi

Gambar 4.3 merupakan rancangan *interface* dari menu dekripsi dimana user menginputkan *key* lalu mengklik tombol nyalakan untuk mengaktifkan kamera pada perangkat yang nantinya kamera tersebut digunakan untuk proses scan *QR Code* dan sistem akan langsung memproses hasil dari proses *scan* tadi.



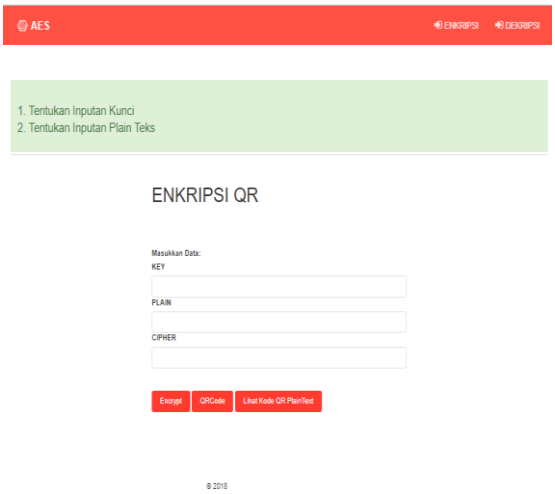
Gambar 4.4 Perancangan antarmuka enkripsi

Gambar 4.4 merupakan rancangan *interface* dari menu enkripsi dimana user mengimputkan *key*, *plaintext* lalu mengklik tombol *encrypt* untuk memulai proses enkripsi dan menklik tombol *QR Code* untuk menampilkan gambar *QR Code* atau lihat kode QR *plaintext* untuk melihat *QR Code* dari *plaintext*.

5. IMPLEMENTASI

5.1 Implementasi Antarmuka

Implementasi *interface* sistem AES ada dua yaitu halaman enkripsi dan dekripsi. Halaman enkripsi merupakan halaman antarmuka bagi user untuk melakukan proses enkripsi pada sistem AES dapat dilihat pada Gambar 5.1 dan Halaman dekripsi merupakan halaman antarmuka bagi user untuk melakukan proses dekripsi pada sistem AES dapat dilihat pada Gambar 5.2.



Gambar 5.1 Tampilan halaman enkripsi

Gambar 5.2 Tampilan halaman dekripsi

5.2 Implementasi Algoritme AES

Implementasi AES untuk enkripsi dan dekripsi pada *QR Code* dengan cara menginputkan *key* dan *plaintext* dirubah menjadi *QR Code*. Lalu sistem akan menjalakan proses enkripsi sebanyak 10 kali putaran yaitu *AddRoundKey, ShiftRows, SubBytes, dan MixColumns* namun pada putaran terakhir tidak dilakukan *MixColumns* dan menghasilkan sebuah *ciphertext* QR *Code*. QR *Code* dapat menampung data a-z, 0-1 dan simbol sehingga dapat memperbanyak variasi data yang akan dienkripsi dan juga memperbanyak variasi *key* sehingga tingkat keamanan lebih tinggi. Sedangkan implementasi dekripsi dengan melakukan *scanning ciphertext* QR *Code* dan memasukkan *key* lalu sistem akan menjalakan proses dekripsi sebanyak 10 kali putaran yaitu *AddRoundKey, InvShiftRows, InvSubBytes, dan InvMixColumns* namun pada putaran terakhir tidak dilakukan *InvMixColumns* dan akan menghasilkan isi pesan asli atau *plaintext*

6. HASIL PENGUJIAN

6.1 Pengujian *Test Vector*

Berikut adalah tabel yang menampilkan hasil dari pengujian *test vector* dari program AES.

Tabel 6.1 hasil pengujian *test vector*

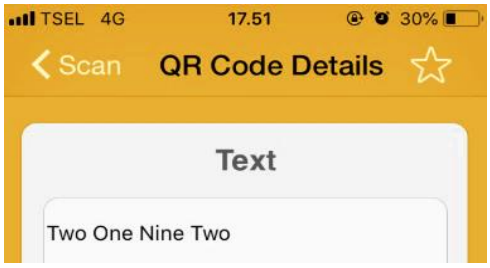
Round	Hasil dari program				Status
	Addroundkey	Subtitusi byte	Shift row	Mix columns	
0	00 3C 6E 47 1F 4E 22 74 0E 08 1B 31 54 59 0B 1A	-	-	-	valid
1	58 15 59 CD 47 B6 D4 39 08 1C E2 DF 8B BA E8 CE	63 EB 9F A0 C0 2F 93 92 AB 30 AF C7 20 CB 2B A2	63 EB 9F A0 2F 93 92 C0 AF C7 AB 30 A2 20 CB 2B	BA 84 E8 1B 75 A4 8D 40 FA 8D 06 7D 7A 32 0E 5D	valid
2	43 0E 09 3D C6 57 08 F8 A9 C0 EB 7F 62 C8 FE 37	6A 59 CB BD A0 4E 48 12 30 9C 98 9E 3D F4 9B 8B	6A 59 CB BD 4E 48 12 A0 98 9E 30 9C 8B 3D F4 9B	15 C9 7F 9D CE 4D 4B C2 89 71 BE 88 65 47 97 CD	valid
3	78 70 99 4B 76 76 3C 39 30 7D 37 34 54 23 5B F1	1A AB 01 27 B4 5B 30 41 D3 BA 39 D2 AA E8 BB 9A	1A AB 01 27 5B 30 41 B4 E9 D2 D3 BA A9 AA E8 BB	AA 65 FA 88 16 0C 05 3A 3D C1 DE 2A B3 4B 5A 0A	valid
4	B1 08 04 E7 CA FC B1 B2 51 54 C9 6C ED E1 D3 20	BC 51 EE B3 38 38 EB 12 04 FF 9A 18 20 26 39 A1	BC 51 EE B3 38 EB 12 38 9A 18 04 FF A1 20 26 39	10 BC D3 F3 D8 94 E0 E0 53 EA 9E 25 24 40 73 7B	valid
5	9B 23 5D 2F 51 5F 1C 38 20 22 BD 91 68 F0 32 56	C8 30 F2 94 74 B0 C8 37 D1 20 DD 50 55 F8 66 B7	C8 30 F2 94 B0 C8 37 74 DD 50 D1 20 B7 55 F8 66	2A 26 8F E9 78 1E 0C 7A 1B A7 6F 0A 5B 62 00 3F	valid
6	14 8F C0 5E 93 A4 60 0F 25 2B 24 92	14 26 4C 15 D1 CF 9C 07 B7 93 7A 81	14 26 4C 15 CF 9C 07 D1 7A 81 B7 93	A9 37 AA F2 AE D8 0C 21 E7 6C B1 9C	valid

	77 EB 40 75	45 8C 23 B1	B1 45 8C 23	F0 FD 67 3B	
7	53 43 4F 85 39 06 0A 52 8E 93 3B 57 5D F8 95 BD	F4 73 BA 58 DC 49 D0 76 3F F1 36 4F F3 9B 09 9D	F4 73 BA 58 49 D0 76 DC 36 4F 3F F1 9D F3 9B 09	9F 37 51 37 AF EC 8C FA 63 39 04 66 4B FB B1 D7	valid
8	66 70 AF A3 25 CE D3 73 3C 5A 0F 13 74 A8 0A 54	ED 1A 84 97 12 6F 67 00 19 DC E2 5B 4C 41 2A 7A	ED 1A 84 97 6F 67 00 12 E2 5B 19 DC 7A 4C 41 2A	E8 8A 4B F5 7475 EE E6 D3 1F 75 58 55 8A 0C 38	valid
9	09 A2 F0 7B 66 D1 FC 3B 8B 9A E6 30 78 65 C4 89	33 51 79 0A 3F 8B 66 8F EB BE 76 7D 92 C2 67 20	33 51 79 0A 8B 66 8F 3F 76 7D EB BE 20 92 C2 67	B6 E7 51 8C 84 88 98 CA 34 60 66 FB E8 D7 70 51	valid
10	29 57 40 1A C3 14 22 02 50 20 99 D7 5F F6 B3 3A	01 3A 8C 21 33 3E B0 E2 3D B8 8E 04 BC 4D 1C A7	01 3A 8C 21 3E B0 E2 33 8E 04 3D B8 A7 BC 4D 1C	-	valid

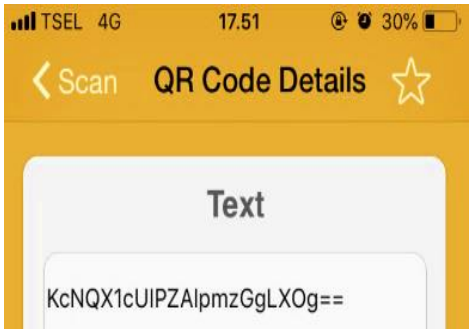
Didapatkan hasil untuk semua round pengujian *test vector* tersebut 100% valid. Sehingga dapat disimpulkan bahwa implementasi algoritme AES berhasil dan sesuai dengan *test vector* nya.

6.2 Pengujian Keamanan

Berikut adalah Hasil dari pengujian keamanan dari QR-Code yang dihasilkan program AES.



Gambar 6.1 Tampilan kode QR *plain text*



Gambar 6.2 Tampilan kode QR *ciphertext*

Dari hasil yang terbaca menunjukan bahwa QR-Code telah dapat menyembunyikan pesan aslinya setelah dienkripsi.

6.3 Pengujian Waktu Enkripsi Dan Dekripsi

Pada pengujian waktu enkripsi dan dekripsi didapatkan hasil waktu enkripsi dan dekripsi kurang lebih sama besar yaitu untuk waktu enkripsi 0.0034 detik dan waktu dekripsi 0.0029 detik.

6.4 Pengujian Fungsional

Untuk pengujian fungsional menggunakan metode black box dengan mencoba seluruh fungsional yang ada pada sistem dan menunjukan hasil yang valid tidak ada *error* pada sistem semua fungsional pada sistem dapat dijalankan tanpa masalah.

6.5 Pengujian *Non*-Fungsional

Untuk pengujian *Non*-Fungsional menggunakan parameter *portability* dan menghasilkan valid. Sistem dapat dijalankan di *browser* Google Chrome, Mozilla Firefox dan opera pada sistem operasi Windows.

7. KESIMPULAN

1) Algoritme AES dapat diterapkan pada QR-*Code*. Algoritme AES akan memberikan aspek *confidentiality*, hal ini dapat dibuktikan dengan pengujian keamanan. Algoritme AES pada proses enkripsi menghasilkan output berupa *ciphertext* berupa karakter tidak jelas yang sulit dipahami dan pada proses dekripsi dilakukan dengan menscan *ciphertext* QR-*Code* dan memasukkan *key* lalu sistem akan

menjalankan proses dekripsi AES dan menghasilkan isi pesan asli atau *plaintext.*

2) Berdasarkan pengujian validasi *plaintext dan ciphertext* pada algoritme AES dapat dibuktikan pada pengujian *test vector* pada algoritme AES. Pada setiap hasil *output* program dari setiap *round* proses enkripsi maupun dekripsi setelah dibandingkan menunjukan hasil yang valid secara keseluruan.

3) Berdasarkan hasil pengujian kinerja pemrosesan enkripsi dan dekripsi algoritme AES pada QR *Code* membutuhkan waktu enkripsi 0.0034 detik dan dekripsi membutuhkan waktu 0.0029 detik.

8. SARAN

Pada penelitian ini terdapat beberapa saran yang dapat digunakan untuk penelitian selanjutnya. Pertama, Objek *QR Code* bisa diganti dengan objek lain misalnya barcode atau gambar. Kedua, AES ini dapat digantikan dengan algoritme enkripsi dan dekripsi lainnya agar dapat mengetahui algoritme mana yang lebih baik untuk objek tertentu. Ketiga, dalam sistem AES ini dapat ditambahkan 2 atau 3 aloritme lagi untuk meningkatkan keamanan dari hasil enkripsi dan dekripsi.

9. DAFTAR PUSTAKA

Ariadi, 2011. *Analisis Dan Perancangan Kode Matriks Dua Dimensi Quick Response (QR) Code*.. [online] Repository.usu.ac.id. Available at: <http://repository.usu.ac.id/handle/123456789/29816> [Accessed 30 June 2018].

Barent, A., 2014. [online] Adamberent.com. Available at: <http://www.adamberent.com/documents/AESbyExample.pdf> [Accessed 2 July 2018].

Daemen, J. and Rijmen, V., 1999. The Rjndael Block Cipher. AES Proposal : Rijndael. *The Rjndael Block Cipher*, [online] Available at: <https://csrc.nist.gov/csrc/media/projects/cryptographic-standards-and-guidelines/documents/aes-development/rijndael-ammended.pdf> [Accessed 30 June 2018].

Denso, A., 2011. [online] Nacs.org. Available at: <http://www.nacs.org/LinkClick.aspx%3Ffileticket%3DD1FpVAvvJuo%253D%26tabid%3D1426%26mid%3D4802> [Accessed 30 June 2018].

Harahap, M., 2016. Analisis Perbandingan Algoritma Kriptografi Klasik Vigenere Cipher Dan One Time Pad. InfoTekJar (Jurnal Nasional Informatika dan Teknologi Jaringan), [online] 1(1), pp.61-64. Available at: <https://jurnal.uisu.ac.id/index.php/infotekjar/article/view/43/30> [Accessed 18 July 2018]

Kavaliro.com. 2014. [online] Available at: <https://kavaliro.com/wp-content/uploads/2014/03/AES.pdf> [Accessed 2 July 2018].

Munir, R., 2006. *Kriptografi*. 1st ed. Bandung: Informatika Bandung.

Sholeh, M. and Muharom, L., 2016. SMART Presensi Menggunakan Qr-Code Dengan Enkripsi Vigenere Cipher. Limits: Journal of Mathematics and Its Applications, 13(2), p.31.

Sumardi, 2017. Studi Model Algoritma Kriptografi Klasik dan Modern. [online] Available at: <http://seminar.uny.ac.id/semnasmatematika/sites/seminar.uny.ac.id.semnasmatematika/files/full/T-37.pdf> [Accessed 17 July 2018].