Implementasi Kriptografi Dengan Metode *Advanced Encryption Standard* (AES) Untuk *Realtime Chat* Berbasis *Mobile* Pada *E-Learning* Politeknik Negeri Lhokseumawe

Rizky F[1*], Anwar[2]

[1,2] *Jurusan Tekniknologi Informasi dan Komputer Politeknik Negeri Lhokseumawe*
Jln. B.Aceh Medan Km.280 Buketrata 24301 INDONESIA

[1*]rizkyfadlyanda@gmail.com

[2]anwarsy@pnl.ac.id

Abstrak— ***E-Learning* merupakan sebuah sistem atau konsep pendidikan yang memanfaatkan ilmu teknologi informasi dan komputer dalam proses belajar mengajarnya.. Untuk mengoptimalkan kinerja *E-Learning*, kebutuhan akan *E-Learning* pun harus diikuti oleh fitur yang memadai, salah satunya adalah fitur *Group Chat* pada tiap mata kuliah. Namun, dikarenakan proses *Chatting* menggunakan koneksi internet, komunikasi yang berlangsung belum tentu aman dari tindakan kejahatan seperti penyadapan atau manipulasi pesan. Oleh karena itu, diperlukan penerapan kriptografi pada komunikasi *Chatting* tersebut. Metode kriptografi yang digunakan adalah algoritma *Advanced Encyption Standard* (AES) dengan panjang kunci sebesar 128-*bit*. Proses komputasi enkripsi yang dilakukan menggunakan kunci yang berbeda-beda pada tiap mata kuliah sehingga membuat *chipertext* untuk kata yang sama menjadi berbeda. Jika terjadi manipulasi pesan, maka akan terdapat pemberitahuan bahwa pesan tersebut telah dimanipulasi keasliannya. Performa kecepatan algoritma AES berpengaruh pada spesifikasi perangkat android yang digunakan dan kinerja CPU. Pengujian kecepatan enkripsi dilakukan dengan rentang panjang pesan dari 16 karakter sampai dengan 1506 karakter untuk lima kali percobaan. Dari pengujian tersebut diperoleh hasil bahwa pada proses enkripsi membutuhkan waktu yang cukup lama jika dibandingkan dengan proses dekripsi. Proses dekripsi membutuhkan waktu rata-rata selama 3,64 ms lebih cepat dari pada proses enkripsi. Kemudian pengukuran efektifitas kriptografi dilakukan menggunakan metode *Avalanche Effect* menghasilkan nilai sebesar 50% untuk pengujian beda *plaintext* dan 49% pada pengujian beda kunci.**
Kata kunci— E-Learning, Kriptografi, Advanced Encryption Standard, AES, Group Chat, Avalanche Effect.

Abstract— **E-Learning is an educational system or concept that utilizes information technology and computers in the learning process. To optimize the performance of E-Learning, the need for E-Learning must also be followed by adequate features, one of which is the Group *Chat* feature in every course. However, because the *Chat* process uses an internet connection, the communication that takes place is not necessarily safe from criminal acts such as wiretapping or message manipulation. Therefore, it is necessary to apply cryptography to the *Chat* communication. The cryptographic method used is the *Advanced Encryption Standard* (AES) algorithm with a key length of 128-bits. The encryption process is carried out using different keys in each course so that it makes the ciphertext for the same word will be different. If a message occurs, there will be a notification that the message has been manipulated. The Encryption speed testing is carried out with a message length range from 16 characters to 1506 characters for five tests. From these tests, it was found that the encryption process takes a long time when compared to the decryption process. The decryption process takes an average of 3.64 ms faster than the encryption process. Then the measurement of the effectiveness of cryptography using the Avalanche Effect method produces a value of 50% for the plaintext difference test and 49% for the key difference test.**
Keywords— E-Learning, Cryptography, Advanced Encryption Standard, AES, Group Chat, Avalanche Effect.

I. PENDAHULUAN

E-Learning merupakan sebuah sistem atau konsep pendidikan yang memanfaatkan ilmu teknologi informasi dan komputer dalam proses belajar mengajar. Telah banyak sekolah atau perguruan tinggi yang telah menerapkan konsep *E-Learning* tersebut, salah satunya adalah Politeknik Negeri Lhokseumawe. Untuk mengoptimalkan kinerja *E-Learning*, kebutuhan akan *E-Learning* pun harus diikuti oleh fitur yang memadai, salah satunya adalah Fitur *Group Chat*. Mahasiswa ataupun dosen yang terhubung ke dalam *Group Chat* dapat memanfaatkannya sebagai forum diskusi, penyebaran informasi, ataupun pemberian arahan terkait materi pembelajaran tanpa dibatasi oleh tempat dan waktu. Namun dikarenakan proses *Chatting* menggunakan koneksi internet, komunikasi yang berlangsung belum tentu aman dari tindakan kejahatan seperti penyadapan atau manipulasi pesan. Oleh karena itu, diperlukan penerapan kriptografi pada komunikasi *Chatting* tersebut.

Penelitian sebelumnya mengenai kriptografi telah dilakukan oleh Deny Adhal pada tahun 2019 dengan judul "Implementasi Algoritma DES (*Data Encryption Standard*) Pada Enkripsi Dan Dekripsi SMS Berbasis Android". Penelitian tersebut memanfaatkan metode enkripsi dan dekripsi dalam membangun sebuah sistem SMS untuk mengamankan pesan atau informasi yang sangat rahasia dari orang-orang yang tidak bertanggung jawab dan tidak berkepentingan, namun pada 1998 terdapat sebuah kelompok yang menamakan dirinya sebagai *Electronic Frontier*

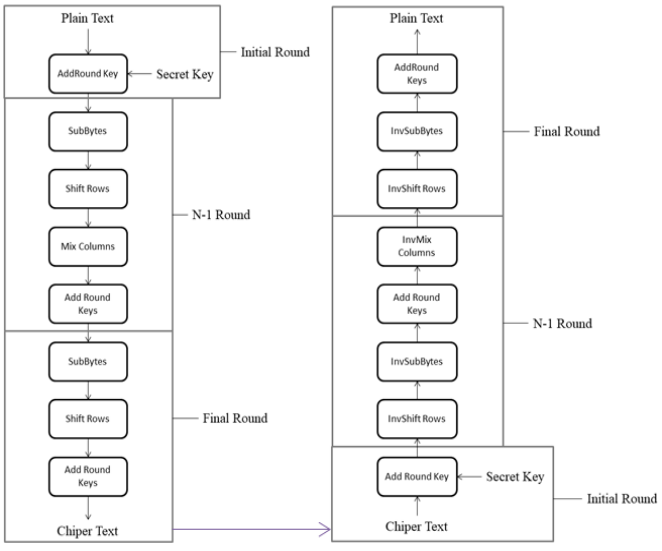
Foundation (EFF) menyatakan telah berhasil memecahkan DES dalam waktu 4-5 hari menggunakan komputer yang dilengkapi dengan *Integrated Circuit Chip* DES *Cracker*. Oleh karena itu, DES dianggap sudah tak aman lagi sehingga diperlukan metode lainnya untuk mengamankan pesan.

Pada *E-Learning* Politeknik Negeri Lhokseumawe belum terdapat fitur *Chat* yang bisa digunakan oleh dosen dan mahasiswa untuk saling berkomunikasi. Dengan latar belakang masalah tersebut, penelitian ini bertujuan untuk mengembangkan sebuah sistem *Group Chat* kelas pada *E-Learning* Politeknik Negeri Lhokseumawe (PNL) berbasis *mobile*. Sistem *Chat* tersebut juga menerapkan metode kriptografi menggunakan metode *Advanced Encryption Standard* (AES). Pesan yang akan dikirimkan (*plaintext*) akan terlebih dahulu melewati proses enkripsi. Hasil dari proses enkripsi tersebut disebut *ciphertext*, *ciphertext* inilah yang kemudian akan di simpan di server sehingga apabila terjadi pembobolan *server* atau penyadapan pesan saat proses trasmisi berlangsung, pesan tersebut akan tetap aman karena akan sulit dan susah untuk dibaca.

II. METODOLOGI PENELITIAN

A. *Metode Mengamankan Pesan*

Metode kriptografi yang digunakan pada penelitian ini untuk mengamankan pesan pada fitur *group Chat E-Learning* PNL adalah *Advanced Encrytion Standard* (AES). AES yang digunakan adalah AES-128 yang menggunakan panjang kunci sebesar 128 *byte* dan putaran sebanyak 10 putaran. Ilustrasi dari proses Enkripsi dan Dekripsi AES dapat dilihat pada gambar 1 berikut:



Gambar 1. Ilustrasi proses enkripsi dan dekripsi AES

Pada gambar 1 tersebut terdapat beberapa proses yang dikerjalan setelah *plaintext* atau pesan dimasukkan yaitu:

1) Initial Round

Pada tahap ini terdapat proses *AddRoundKey* yang melakukan perjumlahan XOR antara *State* awal (*plaintext*) dengan *chipherkey*.

2) Putaran sebanyak 10 kali

Proses awal yang dilakukan pada tiap putaran adalah substitusi *byte* atau disebut *SubBytes* dengan menggunakan tabel substitusi *SubBytes S-box* seperti pada gambar 2 berikut:

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	1	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	4	c7	23	c3	18	96	5	9a	7	12	80	e2	eb	27	b2	75
4	9	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	0	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	2	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
a	e0	32	3a	0a	49	6	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	8
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	3	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Gambar 2. Tabel substitusi untuk transformasi *SubBytes (S-Box)*

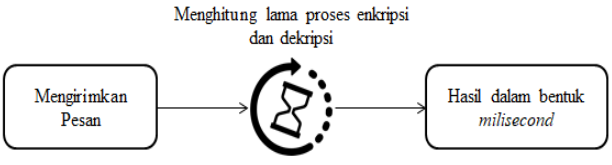
Setelah itu terdapat proses *ShiftRows* yaitu pergeseran baris baris *array State* secara *wrapping*. Selanjutnya *MixColumns* mengacak data di masing masing kolom *array-State* dengan proses perkalian. Kemudian proses *AddRoundKey* melakukan XOR antara *State* sekarang dengan *round key*.

3) Final Round

Terakhir *Final Round* yaitu putaran terakhir yang hanya terdapat beberapa proses tanpa *MixColumns* yaitu *SubBytes*, *ShiftRows*, dan *AddRoundKey*. Sedangkan Deksripsi AES dilakukan dengan cara mengembalikan *chipertext* yang sudah dienkripsi melalui proses transformasi cipher menggunakan *InvShiftRows*, *InvSubBytes*, *InvMixColumns*, dan *AddRoundKey*.

B. *Metode Komparasi Kecepatan Algoritma AES*

Metode untuk menghitung kecepatan algoritma AES pada penelitian ini menggunakan fungsi *build-in* atau fungsi bawaan bahasa pemrograman kotlin yang telah tersedia pada IDE (*Software* untuk pengembangan aplikasi) dalam satuan *milisecond* (ms). Adapaun diagram untuk metode komparasi kecepatan algoritma AES dapat dilihat pada gambar 3 berikut:



Gambar 3. Diagram metode komparasi kecepatan algoritma AES

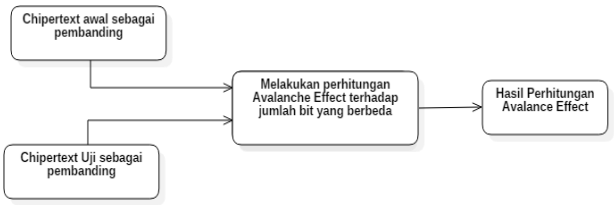
C. *Metode Menghitung Efektivitas Pesan Yang Telah Diamankan*

Metode yang digunakan adalah *Avalance Effect* untuk pengujian perbandingan besar perubahan *bit* yang terjadi pada *chipertext* akibat proses enkripsi AES dengan perhitungan *Avalance Effect* untuk mengetahui nilai persentase efektivitas metode enkripsi tersebut. Tingkat *Avalanche Effect* dapat

dihitung dengan menggunakan rumus pada persamaan 1 berikut:

$$\text{AE} = \frac{\text{Jumlah } bit \text{ yang tergeser pada } chipertext}{\text{Jumlah } bit\ chipertext} \text{ X } 100\% \quad (1)$$

Diagram perhitungan dengan *Avalance Effect* dapat dilihat pada gambar 4 berikut:



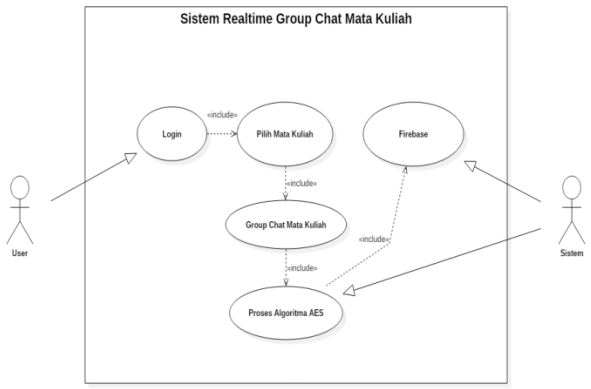
Gambar 4. Diagram perhitungan *Avalanche Effect*

D. Perancangan Sistem

Perancangan sistem merupakan tahapan yang membahas tentang skematis alat serta algoritma pada sistem yang dibangun melalui beberapa diagram.

1) Perancangan Use Case Diagram

Perancangan *Use Case Diagram* pada sistem yang dibangun dapat dilihat pada gambar 5 berikut:

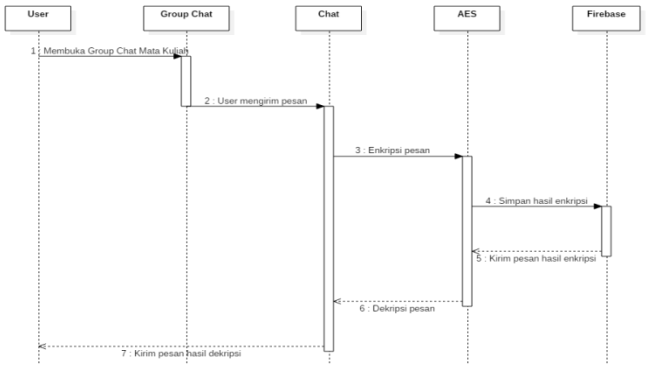


Gambar 5. *Use case diagram* sistem *Realtime Group Chat*

Pada *Diagram* tersebut menunjukkan bahwa user yang telah login akan memiliki beberapa pilihan mata kuliah yang didalamnya terdapat menu *Group Chat* untuk tiap mata kuliah yang dipilih.

2) Perancangan Sequence Diagram

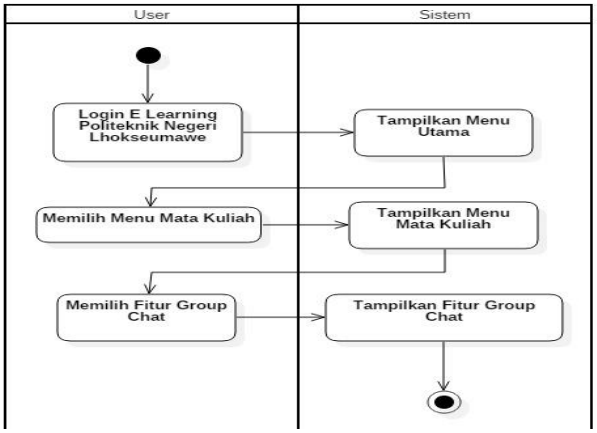
Perancangan *Sequence Diagram* menjelaskan alur dari proses pengiriman pesan dan kapan dilakukan proses enkripsi hingga sampai ke tempat penyimpanan Firebase dan proses deskripsi pesan yang telah di enkripsi menjadi pesan asli yang dapat dibaca. Proses enkripsi dengan menggunakan metode AES dilakukan pada perangkat android setelah users mengirim pesan sebelum disimpan dalam Firebase. Pesan yang tersimpan pada Firebase adalah pesan yang telah berhasil di enkripsi. Setelah itu pesan akan di dekripsi kembali setelah sampai ke penerima pada perangkat androidnya sebelum pesan tersebut dapat dibaca sesuai dengan apa yang dikirim oleh si pengirim. Perancangan *Sequence Diagram* dapat dilihat pada gambar 6 berikut:



Gambar 6. *Sequence Diagram* sistem *Realtime Group Chat*

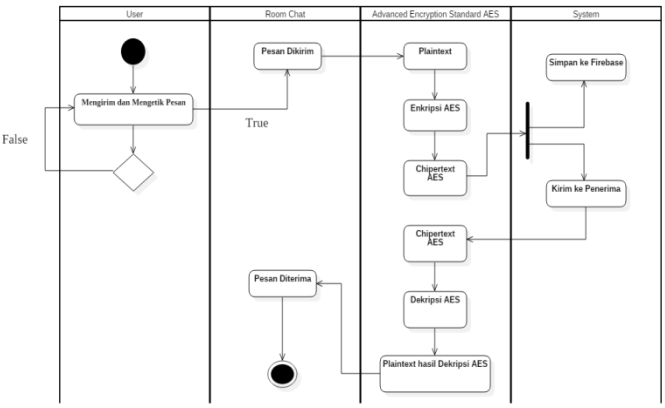
3) Perancangan Activity Diagram

Pada perancangan *Activity Diagram* merupakan *Diagram* yang memodelkan aliran kerja atau *workflow* dari urutan aktifitas dalam suatu proses yang mengacu pada use case *Diagram* yang ada. Berikut adalah penjelasan dari tiap *ActivityDiagram*:



Gambar 7. *Activity Diagram* menampilkan fitur *group chat*

Pada gambar 7 menjelaskan aliran kerja pengguna saat *login* pada aplikasi *E-Learning* Politeknik Negeri Lhokseumawe sampai menampilkan fitur *Group Chat* untuk mata kuliah yang dipilih. Kemudian *Activity Diagram* untuk memulai proses *chatting* dapat dilihat pada gambar 8berikut:



Gambar 8. *Activity Diagram* memulai proses *chatting*

Pada *Activity Diagram* yang ditunjukkan pada gambar 8 menjelaskan aliran kerja pengguna saat memulai proses

chatting, ketika *users* mengirimkan pesan, maka pesan tersebut terlebih dahulu di enkripsi sebelum dikirim ke firebase dengan metode kriptografi *Advanced Encryption Standard* (AES) dan hasil *chipertext* AES akan dikirim ke Firebase. Firebase akan menyimpan data tersebut dan kemudian akan mengirimkannya balik ke si penerima pesan.

III. HASIL DAN PEMBAHASAN

A. *Halaman User Interface*

Sistem *Group Chat* yang dibangun dalam *E-Learning* Politeknik Negeri Lhokseumawe memiliki beberapa antarmuka seperti tampilan *login*, tampilan menu utama,, tampilan *Group Chat*, dan tampilan-tampilan lainnya. Penjelasan penggunaan dari masing masing tampilan tersebut akan diuraikan sebagai berikut.

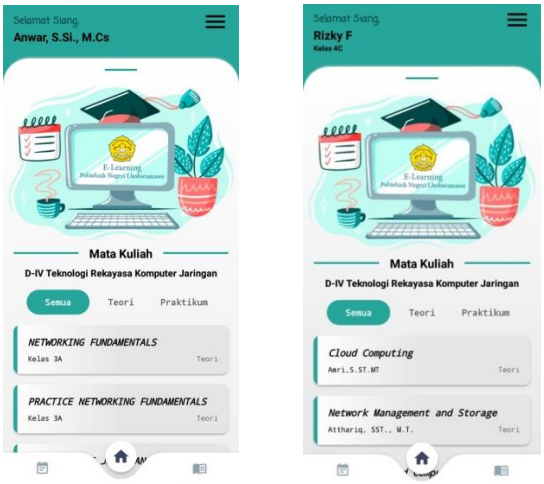
1) *Tampilan Login*

Pada tampilan *login*, dosen dan mahasiswa melakukan proses autentikasi untuk masuk pada aplikasi *E-Learning* berbasis *mobile*. Tampilan login dari aplikasi yang dibangun dapat dilihat pada gambar 9 berikut:

Gambar 9. Tampilan *login*

2) *Tampilan Menu Utama*

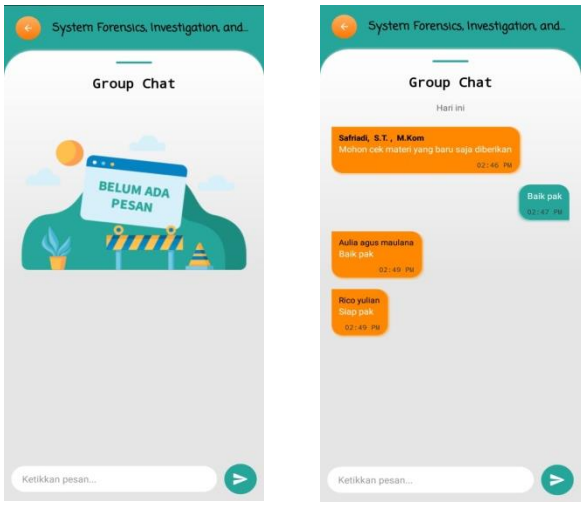
Pada tampilan menu utama, dosen dan mahasiswa dapat memilih mata kuliah yang tersedia sesuai dengan tahun ajaran yang berlangsung seperti pada gambar 10 berikut:



Gambar 10. Tampilan menu utama

3) *Tampilan Group Chat*

Pada tampilan *group chat*, tampilan nama dosen akan tampak lebih tebal dari pada nama mahasiswa untuk membedakan status antara dosen dan mahasiswa. Jika belum terdapat pesan yang dikirimkan maka akan tampil sebuah ilustrasi yang mengatakan *group chat* tersebut belum memiliki sebuah pesan. Tampilan *Group Chat* dapat dilihat pada gambar 11 berikut:



Gambar 11. Tampilan *group chat*

B. *Proses Pengujian Komputasi AES*

Pada pengujian yang dilakukan akan menjelaskan bagaimana proses implementasi dari AES pada fitur *Group Chat* yang dibangun.

1) *Pengujian pengiriman pesan dengan enkripsi AES pada Group Chat*

Pada pengujian ini akan mengirimkan sebuah pesan hanya dengan satu bilangan yaitu huruf “H” seperti pada gambar 12 berikut:

Gambar 12. Mengirimkan pesan ke *group chat*

Huruf "H" memiliki data bilangan hexa dan binari seperti pada tabel I berikut:

TABEL I
DATA HEXA DAN BINARI BILANGAN "H"

Bilangan ASCII	Bilangan Hexa	Bilangan Binari	Ukuran
H	48	01001000	1 *byte*

Dikarenakan pesan yang dikirimkan hanya memiliki ukuran sebesar 1 *byte*, maka pesan tersebut perlu dilakukan padding hingga ukurannya sampai sebesar 16 *byte*. Hal ini dikarenakan AES mewajibkan data yang dienkripsi harus memiliki ukuran sebesar 16 *byte*, tidak boleh lebih dan juga tidak boleh kurang. Hasil padding dari bilangan "H" dapat dilihat pada tabel II berikut:

TABEL II
HASIL PADDING BILANGAN "H"

Bilangan ASCII	Bilangan Hexa Hasil Padding	Bilangan Binari Hasil Padding	Ukuran
H	48 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	01001000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000	16 *byte*

Pada tabel 2 diatas, data yang kurang akan ditambahkan dengan *byte* bernilai 0 sampai menutupi kekurangan *byte* pada pesan yang dikirimkan. *Byte* 0 ini nantinya akan dianggap sebagai ASCII bernilai kosong sehingga pesan yang tampil akan tetap sama yaitu "H". Setelah dilakukan *padding* maka proses enkripsi akan berjalan di *background* aplikasi dan menghasilkan sebuah *chipertext* yang akan disimpan ke dalam *database*. Adapun hasil enkripsi dalam bentuk hexadesimal pada bilangan "H" yang telah dilakukan padding dapat dilihat pada tabel III berikut:

TABEL III
HASIL ENKRIPSI

Tahap Proses	Waktu Saat Pesan Dikirim	Bilangan Hexa
Plaintext awal	21:50:42	48 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Putaran pertama	21:50:42	66 D6 FA 5C F3 41 4B 38 40 07 05 C1 B1 34 4B 50
Putaran kedua	21:50:42	7A A5 87 68 69 1C 1A A4 9A 1E 1B 03 CB 8D BA A7
Putaran ketiga	21:50:42	89 22 F3 5C 9D 03 40 CA E7 FD 74 7F 04 57 5D
Putaran keempat	21:50:42	CD 6D AF 5B 32 43 3F 15 30 63 0D 8E 79 46 1F C1
Putaran kelima	21:50:42	28 4C EA 62 36 A9 38 B1 16 28 DA 61 52 DC E7 09
Putaran keenam	21:50:42	9A 3F 57 8E 3D 2B CA 29 D5 0B 27 76 1B 21 11 6D
Putaran ketujuh	21:50:42	33 E0 D7 06 5D 6C 4C AB 94 32 78 9E 6E EC A2 71
Putaran kedelapan	21:50:42	85 AC D0 3E AA 01 00 9B 13 67 38 86 B5 89 87 6A
Putaran kesembilan	21:50:42	50 99 96 72 94 E1 15 25 F8 C9 B7 BB 20 D7 49 D3
Putaran kesepuluh	21:50:42	D6 C0 61 BB 41 32 2C BA B6 84 36 06 5E BE F8 1E

Dari hasil enkripsi pada tabel 3 didapatkan hasil akhirnya adalah "D6 C0 61 BB 41 32 2C BA B6 84 36 06 5E BE F8 1E", kemudian hasil enkripsi tersebut disimpan kedalam *database* seperti pada gambar 13 berikut:

Gambar 13. Hasil enkripsi pada *database*

Kemudian agar pesan tersebut yang telah menjadi *chipertext* dapat dibaca kembali, *chipertext* tersebut akan didekripsi terlebih dahulu sebelum sampai ke penerima. Hasil dekripsi dari *chipertext* dapat dilihat pada tabel IV berikut:

TABEL IV
HASIL DEKRIPSI

Tahap Proses	Waktu Saat Pesan Dikirim	Bilangan Hexa
Chipertext awal	21:50:42	D6 C0 61 BB 41 32 2C BA B6 84 36 06 5E BE F8 1E
Putaran pertama	21:50:42	97 7C 07 02 AC 85 17 B2 7D A7 70 14 D5 91 63 44
Putaran kedua	21:50:42	C3 50 BC A3 4C 23 3A 6F 22 CE 0E 62 9F E1 29 0B
Putaran ketiga	21:50:42	B8 F1 CC 3C 27 2B 82 19 03 FD 5B A5 AF 75 74 38
Putaran keempat	21:50:42	34 D3 57 01 05 34 94 AA 47 86 87 C8 00 29 07 EF
Putaran kelima	21:50:42	BD 1A D7 78 23 FB C0 39 04 5A 79 59 B6 3C 75 19
Putaran keenam	21:50:42	79 5E 54 4C 4A 94 5B 0D

		74 F2 93 09 D2 A7 7B 92
Putaran ketujuh	21:50:42	DA 9C AF 5C F9 72 F4 45 B8 5D 17 49 1F 06 A2 7B
Putaran kedelapan	21:50:42	33 83 6B 53 0D C5 B3 4A 09 18 2D 07 C8 F6 B3 78
Putaran kesembilan	21:50:42	D2 04 E3 63 C7 23 63 04 04 63 9A 23 20 23 C7 53
Putaran kesepuluh	21:50:42	48 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Pada tabel 4 hasil dekripsi, terlihat bahwa hasil akhir di putaran kesepuluh yaitu “48 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00” telah kembali menjadi *plaintext* awal saat pesan telah melakukan proses *padding*. Jika hexa tersebut diubah menjadi bilangan ASCII maka hanya *byte* 48 saja yang akan berpengaruh menjadi huruf “H”, sedangkan *byte* 0 akan dianggap kosong. Tampilan dari sisi penerima pesan dapat dilihat pada gambar 14 berikut:

Gambar 14. Menerima pesan dari *group chat*

Pada gambar 14 terlihat bahwa pesan yang dikirimkan berhasil tampil tanpa adanya *error* atau kerusakan saat proses dekripsi. Jika terdapat *error* saat melihat pesan yang masuk, maka ada yang salah pada saat proses enkripsi ataupun dekripsinya.

2) Pengujian untuk tiap pesan pada mata kuliah yang berbeda

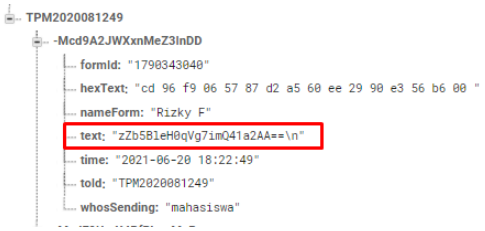
Pada sistem *Group Chat* yang dibangun untuk setiap mata kuliah memiliki kunci yang berbeda sehingga akan menghasilkan sebuah *chipertext* yang berbeda apabila pesan yang dienkripsi mengandung kalimat yang sama. Adapun pengujian yang dilakukan dapat dilihat pada tabel V dengan pesan “Halo apa kabar”.

TABEL V
HASIL PENGUJIAN PADA MATA KULIAH YANG BERBEDA

Mata Kuliah	Chipertext
Practice System Forensics, Investigation, and Response	CD 96 F9 06 57 87 D2 A5 60 EE 29 90 E3 56 B6 00
Practice Network Management and Storage	76 84 00 DC D7 61 96 BC CC FA 4B C7 D8 82 2D F2
Practice Could Computing	CE E6 6D 99 0C 08 E8 0C EF DA D3 1D 36 E4 6B 70
Network Management and Storage	CA 5A 11 F8 39 54 64 60 85 0E C5 A2 33 5E 8B
Cloud Computing	E8 B2 A1 67 C6 EA A2 2D A7 52 56 74 29 E6 2C 22

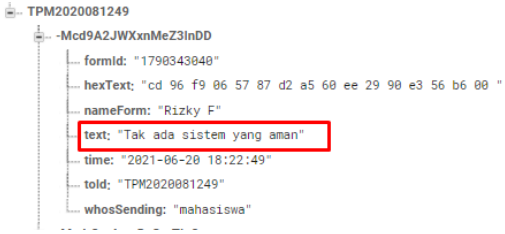
3) Pengujian manipulasi pesan pada Database

Pada aplikasi yang dibangun telah diterapkan sebuah pemberitahuan apabila telah terjadi manipulasi pada pesan di *database*. Terlihat pada gambar 15 pesan yang berada dalam nilai *text* tidak dapat dibaca karena disebabkan oleh hasil enkripsi menggunakan AES yang disebut *chipertext* dalam format *base64*.



Gambar 15. *Chipertext*

Selanjutnya melakukan percobaan manipulasi pesan pada nilai *text* tersebut dengan cara merubah katanya menjadi sebuah kata yang dapat dibaca yaitu “Tak ada sistem yang aman” seperti pada gambar 16 berikut:



Gambar 16. Merubah *Chipertext*

Setelah manipulasi pesan dilakukan, seharusnya proses dekripsi tidak akan dapat berjalan disebabkan oleh tidak sesuainya pola kunci yang diberikan saat proses enkripsi. Oleh karena itu pada pesan yang telah dimanipulasi akan tampil seperti pada gambar 17 berikut:

Gambar 17. Hasil manipulasi pesan

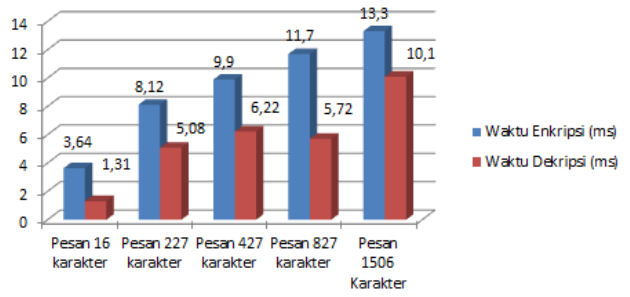
C. Pengujian komparasi kecepatan AES berdasarkan ukuran pesan

Dalam pengujian ini melakukan percobaan menghitung seberapa cepat proses enkripsi dan dekripsi pada AES yang diterapkan berjalan. Rentang panjang kalimat pada pesan yang dikirimkan akan terdapat 5 tahap, tahap pertama akan mengirimkan pesan dimulai dari pesan yang berisi kalimat pendek sebanyak 16 karakter dan yang terpanjang sebanyak maksimal dari pengiriman pesan pada sistem yaitu 1506 karakter. Adapun hasil dari pengujian kecepatan enkripsi dan dekripsi dapat dilihat pada tabel VI berikut:

TABEL VI
HASIL PENGUJIAN KECEPATAN ENKRIPSI DAN DEKRIPSI PADA AES

Panjang Pesan	Proses Waktu		Perbedaan Jarak Waktu Enkripsi dan Dekripsi
	Enkripsi	Dekripsi	
16 karakter	3,64 ms	1,31 ms	2,33 ms
227 karakter	8,12 ms	5,08 ms	3,04 ms
427 karakter	9,90 ms	6,22 ms	3,68 ms
827 karakter	11,70 ms	5,72 ms	5,98 ms
1506 karakter	13,30 ms	10,10 ms	3,20 ms

Berdasarkan pengujian pada tabel 6 yang diterapkan pada 5 data pesan dan menempuh 5 kali pengujian pada masing-masing pesan, diperoleh hasil bahwa pada proses enkripsi membutuhkan waktu yang cukup lama jika dibandingkan dengan proses dekripsi. Proses dekripsi membutuhkan waktu rata-rata selama 3,64 ms lebih cepat dari pada proses enkripsi. Adapun tampilan grafik dari kecepatan enkripsi dan dekripsi dapat dilihat pada gambar 18 berikut:



Gambar 18. Grafik waktu enkripsi dan dekripsi

D. Pengujian Efektivitas Kriptografi AES Pada Group Chat Menggunakan Metode Avalance Effect

Salah satu cara untuk menentukan baik atau tidaknya suatu algoritma kriptografi adalah dengan melihat *Avalanche Effect*-nya. Suatu algoritma kriptografi memenuhi kriteria *Strict Avalanche Criterion* (SAC) apabila rata-rata perubahan *bit* keluaran terhadap berubahnya satu *bit* pada masukan, setidaknya menyentuh 50%. Semakin besar *Avalanche Effect* akan semakin baik algoritma kriptografi tersebut.

Berikut adalah beberapa hasil pengujian pada algoritma *Advanced Encryption Standard* dengan metode *Avalanche Effect* yang ditunjukkan dalam tabel-tabel berikut:

TABEL VII
PLAINTEXT DAN KUNCI AWAL PERBANDINGAN

***Plaintext* awal**	**Kunci**	**Hex *Chipertext* Awal**	**Biner Chipertext Awal**
PoliteknikpnlLSM	ElearningPNL-LSM	4A AF 46 08 2C C8 93 BA 87 91 4A A7 CE 7F 1C FA	01001010 10101111 01000110 00001000 00101100 11001000 10010011 10111010 10000111 10010001 01001010 10100111 11001110 01111111 00011100 11111010

Pada tabel VII diatas merupakan data yang akan menjadi pembanding untuk melakukan pengujian *Avalanche Effect*. Kata *plaintext* awal yang digunakan adalah "PoliteknikpnlLSM" dan kuncinya adalah "ElearningPNL-LSM". Pengujian yang akan dilakukan akan memiliki dua tahap, tahap pertama yaitu pengujian berdasarkan beda *plaintext* dan tahap kedua yaitu pengujian beda kunci. Setiap tahap memiliki tiga kali pengujian dan setiap pengujian hanya mengubah 1 huruf pada kata terakhir. Adapun hasil pengujian beda *plaintext* dengan kunci "ElearningPNL-LSM" dapat dilihat pada tabel VIII berikut:

TABEL VIII
HASIL UJI *AVALANCHE EFFECT* BERDASARKAN BEDA *PLAINTEXT*

***Plaintext* Uji**	**Hex *Chipertext* Uji**	**Biner Chipertext Uji**	**Total *Bit* Beda**	***Avalanche Effect* (%)**
Politeknikpnl LSM	4A AF 46 08 2C C8 93 BA 87 91 4A A7 CE 7F 1C FA	01001010 10101111 01000110 00001000 00101100 11001000 10010011 10111010 10000111 10010001 01001010 10100111 11001110 01111111 00011100 11111010	64	50%
Politeknikpnl LSL	61 74 12 6B 13 AF B6 FC D6 89 7A B9 7A 0B E1 FF	01100001 01110100 00010010 01101011 00010011 10101111 10110110 11111100 11010110 10001001 01111010 10111001 01111010 00001011 11100001 11111111	62	48%
Politeknikpnl LSN	0D C5 9C E4 CD 1C B8 FC D1 C4 4B 58 6F BE B7 06	00011011 10001011 00111001 11001001 10011010 00111001 01110001 11111001 10100011 10001000 10010110 10110000 11011111 01111101 01101110 0000110	67	52%
Rata-Rata *Avalance Effect*				50%

Kemudian pengujian beda kunci dengan *plaintext* "PoliteknikpnlLSM" dapat dilihat pada tabel IX berikut:

TABEL IX
HASIL UJI AVALANCHE EFFECT BERDASARKAN BEDA KUNCI

Kunci	**Hex *Chipertext* Uji**	**Biner Chipertext Uji**	**Total *Bit* Beda**	***Avalanche Effect* (%)**
ElearningPNL-LSA	09 CD CD 1E 02 14 07 FB 68 2A 10 F4 F3 BC 6A 5C	00001001 11001101 11001101 00011110 00000010 00010100 00000111 11111011 01101000 00101010 00010000 11110100 11110011 10111100 01101010 01011100	66	51%
ElearningPNL-LSB	A5 9D CB 16 27 59 83 14 55 D5 87 4D 95 07 5B 31	10100101 10011101 11001011 00010110 00100111 01011001 10000011 00010100 01010101 11010101 10000111 01001101 10010101 00000111 01011011 00110001	64	50%

Elearni ngPNL -LSU	AB F6 AD F6 3F C9 ED 28 CE D0 15 35 6A C7 5C E4	00011011 10001011 00111001 11001001 10011010 00111001 01110001 11111001 10100011 10001000 10010110 10110000 11011111 01111101 01101110 0000110	60	46%
Rata-Rata *Avalance Effect*				49%

Berdasarkan hasil pengujian beda *plaintext* pada tabel VIII menghasilkan rata-rata *Avalanche Effect* sebesar 50%, sedangkan pengujian beda kunci pada tabel IX menghasilkan rata-rata *Avalanche Effect* sebesar 49%. Kedua pengujian tersebut jika digabungkan menghasilkan rata-rata sebesar 49.5% yang jika dibulatkan menjadi sebesar 50%. Dengan ini dapat dikatakan bahwa Algoritma *Advanced Encryption Standard* dengan panjang kunci sebesar 128 *bit* cukup aman terhadap serangan karena mampu menyentuh nilai sebesar 50%. Sampai sekarang pun AES-128 masih belum dapat diretas sehingga AES-128 masih aman untuk digunakan. Jika AES-128 dapat diretas, maka AES-192 dan juga AES-254 pun juga akan dengan mudah dapat diretas karena memilki pola yang sama.

IV. KESIMPULAN

Berdasarkan pembahasan mengenai Implementasi Kriptografi Dengan Metode *Advanced Encryption Standard* (AES) Untuk *Realtime Chat* Berbasis *Mobile* Pada *E-Learning* Politeknik Negeri Lhokseumawe dapat diambil kesimpulan sebagai berikut :

1. Proses komputasi kriptografi pada *Group Chat* yang dilakukan menggunakan kunci yang berbeda-beda pada tiap mata kuliah sehingga membuat *chipertext* untuk kata yang sama menjadi berbeda. Kemudian jika ada pesan yang telah dimanipulasi, pesan tersebut akan tampil dengan pemberitahuan bahwa pesan tersebut telah dimanipulasi keasliannya.
2. Pengujian kecepatan enkripsi dilakukan dengan rentang panjang pesan dari 16 karakter sampai dengan 1506 karakter untuk lima kali percobaan. Dari pengujian tersebut diperoleh hasil bahwa pada proses enkripsi membutuhkan waktu yang cukup lama jika dibandingkan dengan proses dekripsi. Proses dekripsi membutuhkan waktu rata-rata selama 3,64 ms lebih cepat dari pada proses enkripsi.
3. Pengukuran efektifitas kriptografi dilakukan menggunakan metode *Avalanche Effect* menghasilkan nilai sebesar 50% untuk pengujian beda plaintext dan 49% pada pengujian beda kunci.

REFERENSI

[1] Adhar, D. (2019). Implementasi Algoritma DES (Data Encryption Standard) Pada Enkripsi Dan Deskripsi SMS Berbasis Android. *Jurnal Teknik Informatika Kaputama (JTIK)*, *3*(2), 53–60. https://jurnal.kaputama.ac.id/ index.php/JTIK/article/view/185

[2] sir. Muhammad, *Deteksi Usia Tanaman Padi Berdasarkan Indeks Warna,*. Seminar Nasional Teknologi Informasi dan Komunikasi (SNASTIKOM 2013), Medan,2013,Vol 1, hal. 3-145.

[3] Aulia, R., Zakir, A., & Purwanto, D. A. (2018). Penerapan Kombinasi Algoritma *Base64* Dan *Rot47* Untuk Enkripsi *Database* Pasien Rumah Sakit Jiwa Prof. Dr. Muhammad Ildrem. *InfoTekJar (Jurnal Nasional Informatika dan Teknologi Jaringan)*, *2*(2), 146–151. https://doi.org/ 10.30743/ infotekjar. v2i2.300.

[4] Endah. (2020). Fungsi Android Studio. https://metodeku.com/fungsi-android-studio/. Diakses 22 November 2020.

[5] Fahriah, W., & Febrianto, T. (2019). Aplikasi Enkripsi dan Dekripsi *Short Message Service* di Android Menggunakan Metode *Blowfish*. *JISA(Jurnal Informatika dan Sains)*, *2*(1), 1–5. https://doi.org/10.31326/jisa.v2i1.512

[6] Firdaus, Z., & Andri, D. (2019). Implementasi Algoritma *Advanced Encryption Standard* (AES) Sebagai Sistem Pengamanan Data Pengarsipan Pada Perpustakaan Digital Di Puslitbang Geologi Kelautan. *Teknik Informatika-Universitas Komputer Indonesia Jalan Dipatiukur No 112-116 Bandung . 40312*.

[7] Fujimaru, Takagi. (2018) 7 Alasan Kenapa Kita Harus Mulai Belajar Kotlin Untuk Android di 2018. https://www.codepolitan.com/7-alasan-kenapa-kita-harus-mulai-belajar-kotlin-di-2018-5a963b309187d. Diakses 28 November 2020.

[8] Juliarto, Rendi (2021). Apa itu UML? Beserta Pengertian dan Contohnya. https://www.dicoding.com/blog/apa-itu-uml/. Diakses 14 Juli 2021.

[9] Kurniawan, D. (2019). Implementation Of Vigenere Cryptography Algorithm In Lhokseumawe State Polytechnic Storage System. 3(1), 266–271.

[10] Prasasti, I. H., Nanda, A. P., Isnandar, S., Eko, D., & Pramono, H. (2020). Pengembangan Aplikasi *E-Learning* Pada Smk Pelita Bangun Rejo. *1*(1).

[11] Purba, J. A. N., Zebua, T., & Hondro, R. K. (2019). Implementasi Algoritma Paillier *Cryptosystem* Pengamanan Citra Digital Pada Aplikasi *Chat*. *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, *3*(1), 299–306. https://doi.org/10.30865/komik.v3i1.1605.

[12] Ramadhan, A. O., Tolle, H., & Fanani, L. (2018). Pembangunan Modul Penunjang Pembelajaran di Kelas Untuk Aplikasi Brawijaya *Messenger* Dengan Platform *Firebase*. *J-Ptiik.Ub.Ac.Id*, *2*(4), 1630–1637. http://j-ptiik.ub.ac.id.

[13] Suryanto, I., Suhery, C., & Brianorman, Y. (2017). Pengembangan Aplikasi *Chat Messenger* dengan Metode *Advanced Encryption Standard* (AES) pada *Smartphone*. *Jurnal Coding Sistem Komputer Untan*, *03*(2), 1–10.

[14] Wibowo, Dimas Catur. 2019 (Mei). "Apa itu Android? Kenapa *Developer* Memilih Android ?". https://www.dicoding.com/blog/apa-itu-android-kenapa-developer-memilih-android/. Diakses 28 November 2020.

[15] Wijaya, H. (2020). Implementasi Kriptografi AES-128 Untuk Mengamankan URL (*Uniform Resource Locator*) dari SQL *Injection*. *Akademika Jurnal*, *17*(1), 8–13.