**Implementasi Keamanan Chat Realtime Menggunakan Aes-Cbc Dan Base64**

**Implementation Of Realtime Chat Security Using Aes-Cbc And Base64**

 Firdaus Alfajar1 , Mutaqim Akbar2

Program Studi Informatika, Fakultas Teknologi Informasi, Universitas Mercu Buana Yogyakarta, Jl. Wates Km. 10 Yogyakarta 55753, Indonesia

Email: doerikeh1@gmail.com, mutaqin@mercubuana-yogya.ac.id

ABSTRAK

Teknologi pada jaman sekarang banyak dilingkupi oleh kriptografi. Mulai dari percakapan telepon genggam, akses internet, sampai pesan teks telah menggunakan kriptografi. AES-CBC merupakan enkripsi dengan menggunakan kunci dan initial vector, digunakan IV (Initialization Vector) untuk dilakukan XOR dengan blok pertama . Base64 adalah metode pengkodean yang mewakili data biner dalam format string ascii dengan menerjemahkannya kedalam representasi base64. Ada banyak algoritma kriptografi, AES-CBC merupakan enkripsi dengan menggunakan kunci dan initial vector,. Hasil dari penelitian ini adalah algoritma Kriptografi AES-CBC digunakan untuk mengenkripsi pesan dan base64 yang di implementasikan untuk melindungi pesan yang akan di kirim melalui aplikasi chat. Dari penelitian yang dilakukan dapat disimpulkan bahwa hasil berupa 57% pesan yang terdeskripsi memiliki kesamaan dengan isi pesan yang asli karena menggunakan kunci dan initial vector yang sama untuk membuka enkripsi dan,untuk hasil berupa 43% pesan yang terdeskripsi tidak memiliki kesamaan dengan isi pesan yang asli karena menggunakan kunci dan initial vector yang berbeda untuk membuka enkripsi.

**Kata kunci**: Kriptografi, aplikasi chating, AES-CBC, Base64

**ABSTRACT**

Current technology is mostly covered by cryptography. Starting from cell phone conversations, internet access, to text messages, they all have used cryptography. Many have applied it to keep messages secret, but now, many tappers already know the techniques to hack it. Therefore, this study attempts to develop an algorithm that is able to keep text messages secret and then tries to implement it into a chat application.

The results of this research show the AES-CBC cryptography algorithm is used to encrypt messages and base64 is implemented to protect messages to be sent via chat applications.

From the research, it can be concluded that the results show that 57% of the messages encrypted have similarities to the contents of the original message, because they use the same key and initial vector to decrypt and, that 43% of the messages that are decrypted have no similarities to the contents of the original message. because they use different key and initial vector to decrypt.

**Keywords:** cryptography, chatting application*,* AES-CBC, Base64

**1. PENDAHULUAN**

**IMPLEMENTASI KEAMANAN CHAT REALTIME MENGGUNAKAN AES-CBC DAN BASE 64**

Teknologi informasi dan telekomunikasi saat ini berkembang sangat pesat dan memberikan banyak pengaruh bagi kehidupan manusia. Hal yang paling jelas yang dialami saat ini adalah perkembangan jaringan *internet* yang sangat membantu manusia melakukan banyak kegiatan seperti bertukar data dan informasi dengan orang lain melalui *internet*. Namun seiring dengan meluasnya penggunaan jaringan *internet*, pengiriman informasi pun semakin rentan terhadap penyadapan yangdapat mengubah integritas data. Aplikasi *Chatting* adalah salah satu media komunikasi yang sering digunakan untuk menyampaikan pesan. Pada kepentingan atau tujuan tertentu seseorang ingin mengirim pesan yang isi pesannya tidak ingin diketahui oleh orang lain selain si penerima pesan yang dituju karena isi pesan tersebut bersifat sangat rahasia atau pribadi. Tentu kejahatan dalam dunia maya merupakan hal yang sangat merugikan baik bagi pengguna *internet* maupun penyedia jasa *internet*. Namun kenyataannya banyak kasus pencurian data atau penyadapan data yang sangat rahasia bisa dibobol oleh pihak yang tidak bertanggung jawab yang biasa dikenal sebagai *cybercrime*. Untuk mengamankan data penting yang berupa informasi tersebut dibutuhkan suatu kriptografi.

Salah satu sarana *teknologi* yang digunakan dalam mengirim pesan yaitu aplikasi *chatting*, mayoritas pengguna *internet* berkomonikasi menggunakan chatting yang memungkinkan pengguna saling terhubung untuk melakukan komunikasi di tempat yang berbeda dengan relatif waktu yang singkat. Pada perkembangannya applikasi *chatting* tidak hanya digunakan untuk mengirim pesan berupa teks tapi dapat berupa gambar, pesan suara dan lainnya .

berbagai permasalahan tersebut dapat diatasi dengan proses enkripsi, yang cukup dikenal adalah dengan metode enkripsi *AES* (*Advanced Encryption Standar*) dan *base 64*. metode enkripsi ini akan memberikan *private key*. yang digunakan dalam proses enkripsi dan deskripsi, pada penelitian ini algoritma *AES-CBC* akan di implementasikan pada salah satu teknik berkomunikasi yaitu *chatting*. sehingga diharapkan implementasi algoritma *AES-CBC* ini bisa menjadi salah satu cara mengamankan pesan dalam *chatting*.

**2. TINJAUAN PUSTAKA**

**2.1 Chatting**

*chatting* adalah menghubungkan dua orang atau lebih tapi terhubung melalui *internet*. memungkinkan untuk berkomunikasi secara langsung di tempat yang berbeda secara *realtime* yang berupa pesan teks.

**2.2 Kriptografi**

Kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Dalam ilmu kriptografi, terdapat dua buah proses yaitu melakukan enkripsi dan dekripsi. Pesan yang akan diekripsi disebut sebagai plaintext (teks biasa). Disebut demikian karena informasi ini dengan mudah dapat dibaca dan dipahami oleh siapa saja. Algoritma yang dipakai untuk mengenkripsi dan mendekripsi sebuah plaintext melibatkan penggunaan suatu bentuk kunci. Pesan plaintext yang telah dienkripsi (atau dikodekan) dikenal sebagai ciphertext (teks sandi). Di dalam kriptografi kita akan sering menemukan berbagai istilah atau terminology. Beberapa istilah yang harus diketahui yaitu.

1. Pesan, *Plaintext*, dan *Ciphertext*

Pesan (*message*) adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah *plaintext* atau teks-jelas (*cleartext*). Pesan dapat berupa data atau informasi yang dikirim (melalui kurir, saluran telekomunikasi, dsb) atau yang disimpan di dalam media perekaman (kertas, *storage*, dsb). Pesan yang tersimpan tidak hanya berupa teks, tetapi juga dapat berbentuk citra (*image*), suara/bunyi (*audio*), dan video, atau berkas biner lainnya.

1. Pengirim dan Penerima

Komunikasi data melibatkan pertukaran pesan antara dua entitas. Pengirim(*sender*) adalah entitas yang mengirim pesan kepada entitas lainnya. Penerima (*receiver*) adalah entitas yang menerima pesan. Entitas di sini dapat berupa orang, mesin (komputer), kartu kredit, dan sebagainya. Jadi, orang bisa bertukar pesan dengan orang lainnya. Pengirim tentu menginginkan pesan dapat dikirim secara aman, yaitu ia yakin bahwa pihak lain tidak dapat membaca isi pesan yang ia kirim. Solusinya adalah dengan cara menyandikan pesan menjadi *ciphertetx*.

1. Enkripsi dan dekripsi

Proses menyandikan *plaintext* menjadi *ciphertext* disebut enkripsi (*encryption*) atau *enciphering* (standard nama menurut ISO 7498-2). Sedangkan proses mengembalikan *ciphertext* menjadi *plaintext* semula dinamakan dekripsi (*decryption*) atau *deciphering* (standard nama menurut ISO 7498-2). *Enkripsi* dan *dekripsi* dapat diterapkan baik pada pesan yang dikirim maupun pada pesan tersimpan. Istilah *encryption* *of data in motion* mengacu pada enkripsi pesan yang ditransmisikan melalui saluran komunikasi, sedangkan istilah *encryption* *of data at-rest* mengacu pada *enkripsi* dokumen yang disimpan di dalam *storage*.

1. *Cipher* dan kunci

Algoritma *kriptogarfi* disebut juga *cipher*, yaitu aturan untuk enkripsi dan dekripsi, atau fungsi matematika yang digunakan untuk enkripsi dan dekripsi. Beberapa *cipher* memerlukan algoritma yang berbeda untuk enkripsi dan dekripsi. Konsep matematis yang mendasari algoritma *kriptografi* adalah relasi antara dua buah himpunan yang berisi elemen-elemen *plaintext* dan himpunan yang berisi *ciphertext*. Enkripsi dan dekripsi merupakan fungsi yang memetakan elemen-elemen antara dua himpunan tersebut. Misalkan P menyatakan *plaintext* dan C menyatakan *cipherteks*, maka :

E(P) = C 🢡 fungsi enkripsi E memetakan P ke C

D(C) = P 🢡 fungsi dekripsi D memetakan C ke P

Karena proses enkripsi kemudian dekripsi mengembalikan pesan ke pesan asal, maka persamaan D(E(P)) = P harus benar.

**2.3 Anvanced Encryption Standard (AES)**

*AES* merupakan algoritma Rijndael yang ditemukan oleh Dr.Vincent Rijmen dan Dr.Joan Daemen merupakan algoritma simetri dan *cipher* blok. Dengan demikian algoritma ini menggunakan kunci yang sama pada saat enkripsi dan deskripsi serta *input* dan *output* berupa blok dengan jumlah bit tertentu. Algoritma *AES* menggunakan substitusi, permutasi, dan sejumlah putaran yang dikenakan pada tiap blok yang akan dienkripsi/dekripsi. Untuk setiap putarannya, *AES* menggunakan kunci yang berbeda. Kunci setiap putaran disebut *round key*. Tetapi tidak seperti *DES* yang berorientasi bit, *AES* beroperasi dalam orientasi *byte* sehingga memungkinkan untuk implementasi algoritma yang efisien ke dalam *software* dan *hardware.* Ukuran blok untuk algoritma *AES* adalah 128 bit, 192 bit, dan 126 bit dari ketiga versi AES tersebut terdapat perbedaan pada jumlah *key* dan putaranya seperti pada tabel berikut:

*Tabel. 1 AES*

|  |  |  |  |
| --- | --- | --- | --- |
| Versi AES | Jumlah Key | Jumlah Blok | Jumlah Putaran |
| AES-126 | 4 | 4 | 10 |
| AES-192 | 6 | 4 | 12 |
| AES-256 | 8 | 4 | 14 |

**2.4 AES-CBC**

Mode operasi CBC ditemukan oleh IBM pada tahun 1976. Pada mode ini, tiap blok dari *plaintext* dilakukan *XOR* dengan hasil *ciphertext* dari blok sebelumnya yang kemudian dilakukan enkripsi. Dengan cara ini, tiap *ciphertext* dari masing-masing blok akan tergantung pada seluruh hasil *ciphertext* dari blok-blok sebelumnya. Selain itu, untuk membuat tiap pesan menjadi unik, digunakan *IV* (*Initialization Vector*) untuk dilakukan *XOR* dengan blok pertama

Jika blok pertama memiliki indeks 1, maka rumus matematis untuk enkripsi pada mode CBC sebagi berikut :

$$C\_{i}= E\_{k}\left(P\_{i}⊕ C\_{i-1}\right), C\_{0}=IV$$

sedangkan rumus matematis untuk dekripsi pada mode CBC Sebagai Berikut:

$$P\_{i}= D\_{k}\left(C\_{1}\right)⊕C\_{i-1,}C\_{0}=IV$$

$C\_{i}$ : *Ciphertext* pada blok i

$P\_{i}$: *Plaintext* pada blok i

$E\_{k}(…)$: Funsi enkripsi yang digunakan

$D\_{k}(…)$:Fungsi Dekripsi yang digunakan

$IV$: *Initialzation Vector*

* 1. **Base64**

Transformasi Base64 merupakan salah satu algoritma untuk *encoding* dan *decoding* suatu data ke dalam format *ASCII*, yang didasarkan pada bilangan dasar 64 atau bisa dikatakan sebagai salah satu metode yang digunakan untuk melakukan *encoding* (penyandian) terhadap data *binary*. Karakter yang dihasilkan pada transformasi Base64 ini terdiri dari A….Z, a….z dan 0….9, serta ditambah dengan dua karakter terakhir yang bersimbol yaitu + dan / serta satu buah karakter sama dengan (=) yang digunakan untuk penyesuaian dan menggenapkan data *binary* atau istilahnya disebut sebagai pengisi pad. Karakter simbol yang akan dihasilkan akan tergantung dari proses algoritma yang berjalan. *Kriptografi* transformasi Base64 banyak digunakan di dunia *internet* sebagai media data format untuk mengirimkan data, ini dikarenakan hasil dari *Base64* berupa *plaintext*, maka data ini akan jauh lebih mudah dikirim, dibandingkan dengan format data yang berupa *binary*. Dalam implementasinya beberapa contoh dalamtransformasi *Base64*, yang antara lain adalah sebagai berikut.

1. *PEM*

*PEM (Privacy-Enhaced Mail)* adalah protocol pertama dengan teknik *Base64* yang didasarkan pada *RFC 989*, yang terdiri dari 7 karakter (7-bit) yang digunakan pada *SMTP* dalam transfer data tapi untuk sekarang *PEM* sudah tidak menggunakan *RFC 989* tapi sudah di ganti dengan *RFC 1421* yang menggukana karakter A….Z, a….z, 0….9.

1. *MIME*

*MIME (Multi Purpose Mail Extension)* didasarkan pada *RFC 2045.* Teknik *encoding* Base64MIME, mempunyai konsep yang berdasarkan *RFC 1421* versi *PEM*. Sedangkan *MIME* diakhiri dengan *padding* “=” pada hasil akhir *encoding*.

1. *UTF-7*

*UTF-7* didasarkan pada *RFC 2152*, yang umumnya disebut “*MODIFICATION BASE*” UTF-7 menggunakan karakter *MIME*, tidak memakai *padding*”=”, karakter “=” sebagian digunakan untuk *encoding*

*Tabel. 2 Base64*

|  |
| --- |
| Base Encoding Table |
| Value | Char | **Value** | Char | **Value** | Char | **Value** | Char |
| 0 | A | **16** | Q | **32** | G | **48** | W |
| 1 | B | **17** | R | **33** | H | **49** | X |
| 2 | C | **18** | S | **34** | I | **50** | Z |
| 3 | D | **19** | T | **35** | J | **51** | Y |
| 4 | E | **20** | U | **36** | K | **52** | 1 |
| 5 | F | **21** | V | **37** | L | **53** | 2 |
| 6 | G | **22** | W | **38** | M | **54** | 3 |
| 7 | H | **23** | X | **39** | N | **55** | 4 |
| 8 | I | **24** | Y | **40** | O | **56** | 5 |
| 9 | J | **25** | Z | **41** | P | **57** | 6 |
| 10 | K | **26** | A | **42** | Q | **58** | 7 |
| 11 | L | **27** | B | **43** | R | **59** | 8 |
| 12 | M | **28** | C | **44** | S | **61** | 9 |
| 13 | N | **29** | D | **45** | T | **62** | - |
| 14 | O | **30** | E | **46** | U | **63** | \_ |
| 15 | P | **31** | F | **47** | V | **pad** | = |

Teknik enkripsi base64 sebetulnya sangat sederhana, jika terdapat sebuah (*string*) *bytes* yang akan disandikan kedalam algoritma *base64* maka tahapanya yaitu:

1. Pecahan *string* tersebut ke per-3 *bytes*.
2. Gabungkan 3 *bytes* menjadi 24 bit. Dengan catatan 1 *bytes* = 8 bit, sehingga 3 x 8 = 24 bit.
3. Lalu 24 bit yang disimpan di-*buffer* ( disatukan ) dipecah-pecah menjadi 6 bit, maka akan menghasilkan 4 pecahan.
4. Masing-masing pecahan di ubah ke dalam desimal, dimana maksimal nilai 6 bit adalah 63.
5. Terakhir, jadikan nilai-nilai desimal tersebut menjadi *index* untuk memilih maksimal *index* ke 64 atau karakter ke 63 dari penyusunan base64

Dan seterusnya hingga akhir string *bytes* akan mengalami konversi. Apabila dalam proses *encoding* terdapat sisa pembagi. Maka tambahkan katarter pad(=) sebagai penggenap sisa tersebut oleh karena itu, terkadang pada base 64 akan muncul satu atau dua karakter(=).

**3. METODOLOGI PENELITIAN**

*dalam penelitian pembuatan aplikasi chat realtime ini kita akan menggunakan ADCT (Analisis, Design, Coding, Testing). Dengan Langkah-langkah sebagai berikut:*

**3.1 Analisis**

Analisis kebutuhan fungsional adalah kebutuhan yang berisi cara-cara apa saja yang akan dilakukan oleh *system analisis* kebutuhan fungsional terdiri dari:

1. *plaintext* Dapat Terenkripsi menjadi bagian-bagian blok 126 *bytes AES CB*C
2. hasil enkripsi tergantung dari hasil blok-blok sebelum nya
3. enkrpisi akan dirubah ke menjadi *binary* yang diakhiri symbol “=” oleh Base64
4. setiap blok berorientasi *XOR*
5. Enkripsi *plaintext* *AES CBC* dan *Base64* menurut *Key*
6. Dekripsi *plaintext* *AES CBC* dan *Base64* dengan *Key*

**3.2 Desaign Sistem**



*Gambar. 1 Usecase Diagram Chatting*

**3.2.1 Design Interface**

Rancangan desain antar muka atau desain *interface* adalah suatu perancangan antar muka yang bertujuan untuk menghubungkan antara pengguna dengan sistem operasi sehingga komputer dapat di operasikan, atau bisa dibuat sebagai sebuah proses dari pengguna untuk dapat berkomunikasi dengan komputer melalui fitur-fitur yang terdapat didalamnya.



*Gambar. 2 Halaman Chatting*

**3.3 Coding**

Dalam pembuatan aplikasi *chat* ini menggunakan bahasa pemograman dan *library* sebagai berikut:

**3.3.1 Python3**

Pada penelitian ini Bahasa pemograman nya menggunakan python karena mudah di gunakan dan di dukung oleh banyak *library* seperti:

1. *library numpy* yang akan di gunakan untuk inisialisasi *array*
2. *liblary* *Django* di gunnakan sebagai *framework* yang menghubungka antara python dan html,
3. *liblary Django* *channel* di gunnakan untuk menghubungkan antara *client* dan *server*,
4. *liblary* *Django rest framework* di gunnakan untuk *API* pada *Django*,
5. *liblary Django* *mirage* di gunakan untuk mengenkripsi data saat dikirim oleh *client* dan akan di deskripsi saat akan di tampilkan ke *client* lagi.

**3.3.2 Html**

Pada penelitian ini program yang di gunnakan untuk membuat *website* adalah html karena mudah digunakan

**3.3.3 CSS3**

Pada penelitian ini program yang di gunnakan untuk membuat *website* adalah css karena mudah dan banyak di dukung oleh *liblary* seperti *boostrap* yang nanti akan di gunakan untuk mendesign *website*

**3.3.4 Javascript 1.8.5**

Pada penelitian ini program yang di gunnakan untuk membuat *website* adalah *javascript* diggunakan untuk mengoptimalkan aplikasi *chat*.

**3.4 Testing**

Setelah sistem selesai di bangun, maka harus di uji apakah sistem dapat berjalan baik, pengujian akan dilakukan dengan melakukan langkah-langkah:

1. Melakukan pengujian menggunakan *black-box testing*
2. Melakukan Enkripsi sebanyak 1 lembar
3. Melakukan pengujian menggunakan *encryptor online* untuk membuka enkripsi : <https://www.online-toolz.com/tools/text-encryption-decryption.php>

**4. PEMBAHASAN**

Berikut adalah hasil penelitian yang sudah dilakukan. Ada beberapa hal yang dihasilkan dari penelitian ini antara lain metode yang sudah dikembangkan dan aplikasi chatting yang bertujuan untuk mengimplementasikan metode yang sudah dikembangkan

**4.1 Pengujian Sistem**

Pengujian sistem enkripi dengan menggunakan metode AES CBC dan base64 ini dilakukan dengan menggunakan kunci dan *initial vector* yang berbeda yang digunnakan untuk menguji apakah pesan tersebut dapat di baca, Namun tidak menutup kemungkinan pada kondisi tertentu akan gagal. Berikut detail yang dihasilkan.

**4.1.1 Black Box Testing**

Pengujian *Black Box Testing* ini ditujukan untuk melatih keseluruhan unit fungsional dari perangkat lunak agar perangkat lunak dapat bekerja dengan baik .

*Tabel. 3 Black Bos Testing*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Kasus Uji | Skenario | Hasil yang di dapat | Keterangan |
| 1 | User dapat mengirim pesan | Pesan dapat terkirim | System dapat mengirim pesan ke user | berhasil |
| 2 | Melakukan enkripsi | Enkripsi dengan kunci yang sudah di tentukan | System dapat melakukan enkripsi dengan benar | Berhasil |
| 3 | Melakukan deskripsi | deskripsi dengan kunci yang sudah di tentukan | System dapat melakukan deskripsi dengan benar | berhasil |

**4.1.2 Pengujian Enkripsi**

*plaintext* yang terenkripsi akan dibuka menggunakan kunci dan *initial Vector* yang sama maupun tidak sama agar mendapatkan hasil apakah plaintext yang sudah di enkripsi dapat dibuka atau tidak

*Tabel. 4 Pengujian AES-CBC dan Base64*

|  |  |  |  |
| --- | --- | --- | --- |
| No | Panjang | enkripsi | Deskripsi |
| 1 | 197 Kerakter | berhasil | Berhasil |
| 2 | 303 karakter | berhasil | Tidak berhasil |
| 3 | 516 karakter | berhasil | berhasil |
| 4 | 327 Karakter | Berhasil | Tidak Berhasil |
| 5 | 261 Karakter | Berhasil | Tidak berhasil  |
| 6 | 426karakter | Berhasil | Berhasil |
| 7 | 130 karakter | Berhasil | Tidak berhasil |
| 8 | 97 karakter  | Berhasil | Tidak berhasil |
| 9 | 98 karakter | Berhasil | berhasil |
| 10 | 114 karakter | Berhasil | berhasil |
| 11 | 325 karakter | berhasil | berhasil |
| 12 | 533 karakter | Berhasil | Tidak berhasil |
| 13 | 599 karakter | berhasil | berhasil |
| 14 | 880 karakter | berhasil | Tidak berhasil |
| 15 | 339 karakter | Berhasil | berhasil |
| 16 | 339 karakter | Berhasil | berhasil |
| 17 | 310 karakter | berhasil | Tidak berhasil |
| 18 | 280 karakter | berhasil | berhasil |
| 19 | 272 karakter | berhasil | berhasil |
| 20 | 210 karakter | berhasil | Tidak berhasil |
| 21 | 259 karakter | berhasil | berhasil |
| 22 | 332 karakter | berhasil | berhasil |
| 23 | 115 karakter | berhasil | Tidak berhasil |
| 24 | 155 karakter | berhasil | berhasil |
| 25 | 154 karakter | berhasil | Tidak berhasil |
| 26 | 88 karakter | berhasil | berhasil |
| 27 | 97 karakter | berhasil | berhasil |
| 28 | 191 karakter | berhasil | berhasil |
| 29 | 275 karakter | berhasil | Tidak berhasil |
| 30 | 245 karakter | berhasil | Tidak berhasil |

Data yang berhasil dalam pengujian di hitung dengan rumus sebagai berikut $unjuk kerja= \frac{berhasil}{total pengujian} \*100$

$Unjuk Kerja= \frac{17}{30} \*100$ yang akan mendapatkan hasil berupa 57% pesan yang terdeskripsi memiliki kesamaan dengan isi pesan yang asli karena menggunakan kunci *dan initial vector* yang sama untuk membuka enkripsi

Data yang tidak berhasil dalam pengujian di hitung dengan rumus sebagai berikut $unjuk kerja= \frac{Tidak Berhasil}{total pengujian} \*100$

$Unjuk Kerja= \frac{13}{30} \*100$ yang akan mendapatkan hasil berupa 43% pesan yang terdeskripsi tidak memiliki kesamaan dengan isi pesan yang asli karena menggunakan kunci dan *initial vector* yang berbeda untuk membuka enkripsi

**4. KESIMPULAN**

Hasil dari pengembangan dan implementasi algotitma AES CBC dan Base64r ini, ada beberapa kesimpulan yang didapat diantaranya:

1. Aplikasi chating mampu mengimplementasikan algoritma kriptografi. Hasil pengujian 30 dari 30 pesan enkripsi maka membuktikan 57% pesan yang terdeskripsi memiliki kesamaan dengan pesan asli karena menggunakan kunci dan initial vector yang sama untuk membuka enkripsi
2. Hasil pengujian dari 30 dari 30 maka membuktikan 43% pesan yang terdekripsi tidak memiliki kesamaan dengan pesan asli karena menggunakan kunci dan initial vector yang berbeda untuk membuka enkripsi

**5. UCAPAN TERIMA KASIH**

Terima kasih untuk Mu taqim Akbar S.Kom, M.T dan A. Sidiq Purnomo, S.Kom, M,Eng yang telah meluangkan waktu untuk memberikan dukungan, bimbingan, motivasi, dan arahan dalam menyelesaikan jurnal rekrusif ini.

**DAFTAR PUSTAKA**

astuti, f. i., kridalaksana, a. h., & Pabokory, F. n. (2015 ). Informatika Mulawarman. *IMPLEMENTASI KRIPTOGRAFI PENGAMANAN DATA PADA PESAN TEKS, ISI FILE DOKUMEN, DAN FILE DOKUMEN MENGGUNAKAN ALGORITMA ADVANCED ENCRYPTION STANDARD*, 20-31.

Herianto, I. (n.d.). Analisis Perbandingan Performasi Algoritma Advance Encryption Standard (AES) dan twofish. 2015.

herry, kridaklasana, a. h., & arifin, z. (2016). Prosiding Seminar Ilmu Komputer dan Teknologi Informasi Vol. 1, No. 1,. *KRIPTOGRAFI AES MODE CBC PADA CITRA DIGITAL BERBASIS ANDROID*, 45-52 ISSN:2540 – 7902 .

mulyana, d. i. (2016). Jurnal CKI On SPOT, Vol. 9. *KAJIAN PENERAPAN ENCODE DATA DENGAN BASE64 PADA PEMROGRAMAN PHP*, 47-52 ISSN: 1979-7044

Nugraha, A. P., & E. G. (2016). No. 2 2016 : Jurnal Algoritma. *PENERAPAN KRIPTOGRAFI BASE64 UNTUK KEAMANAN URL (UNIFORM RESOURCE LOCATOR) WEBSITE DARI SERANGAN SQL INJECTION*, 491-498.

Nugroho, A. Y. (2015). PEMBUATAN APLIKASI KRIPTOGRAFI ALGORITMA BASE 64 MENGGUNAKAN PHP UNTUK MENGAMANKAN DATA TEXT. *Nasional Informatika*, 134-139.

P.U.K, G. G., & Erlansha, A. (2016). IMPLEMENTASI METODE ADVANCED ENCRYPTION STANDARD (AES) DAN MESSAGE DIGEST 5 (MD5) PADA ENKRIPSI DOKUMEN. *Rekursif, Vol. 4 No. 3* , 277-287.

Qutsiah, S. A., Sophan, K. M., & Hendrawan, Y. F. (2016). APLIKASI PEMBELAJARAN MATEMATIKA DASAR BANGUN DATAR MENGGUNAKAN PYTHON PADA PERANGKAT BERGERAK. *SCAN VOL. XI NOMOR 3*, 13-22.

somya, r. (2018 Vol.4). Khasanah informatika. *Perancangan Aplikasi Chatting Berbasis Web di PT. Pura Barutama Kudus Menggunakan Socket.IO dan Framework Foundation*, 8-15.

Susiliato, C. (2018). *DESAIN ENKRIPSI DATA MENGGUNAKAN ALGORITMA AES 128*.

tulloh, A. R., Permanasari, Y., & Harahap, E. (2016). Jurnal Matematika UNISBA VOL 15. *Kriptografi Advanced Encryption Standard (AES)*, 7-14 ISSN: 1412-5056.