

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

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. Untuk mendapatkan hasil dari penelitian ini akan dilakukan proses enkripsi dan deskripsi text sebanyak 30 kali yang akan menggunakan kunci dan initial vector yang berbeda untuk menguji pesan yang sudah di enkripsi sesuai dengan text awal saat di deskripsi. 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 chatting, AES-CBC, Base64

ABSTRACT

Chatting application is a communication medium that is often used to convey messages. For certain interests or purposes, someone wants to send a message whose contents do not want anyone other than the intended recipient to know because the contents of the message are very confidential or private. To get results from this study will be done the encryption process and text description 30 times that will use different keys and initial vectors to test messages that are already encrypted according to the initial text when in the description. 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

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 yang dapat 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* berkomunikasi menggunakan *chatting* yang memungkinkan pengguna saling terhubung untuk melakukan komunikasi di tempat yang berbeda dengan relatif waktu yang singkat. Pada perkembangannya aplikasi *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 dienkrpsi (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.

2. 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 *ciphertext*.

3. 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*.

4. *Cipher* dan kunci

Algoritma *kriptografi* 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 *ciphertexts*, maka:

$E(P) = C \Rightarrow$ fungsi enkripsi E memetakan P ke C

$D(C) = P \Rightarrow$ 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 Advanced 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 256-bit dari ketiga versi *AES* tersebut terdapat perbedaan pada jumlah *key* dan putarannya.

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 sebagai berikut:

$$C_i = E_k(P_i \oplus C_{i-1}), C_0 = IV$$

sedangkan rumus matematis untuk dekripsi pada mode CBC sebagai berikut:

$$P_i = D_k(C_i) \oplus C_{i-1}, C_0 = IV$$

C_i : *Ciphertext* pada blok i

P_i : *Plaintext* pada blok i

$E_k(\dots)$: Fungsi enkripsi yang digunakan

$D_k(\dots)$: Fungsi Dekripsi yang digunakan

IV : *Initialization Vector*

2.4 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 dalam transformasi *Base64*, yang antara lain adalah sebagai berikut.

1. PEM

PEM (*Privacy-Enhanced 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 menggunakan karakter A...Z, a...z, 0...9.

2. MIME

MIME (Multi Purpose Mail Extension) didasarkan pada *RFC 2045*. Teknik *encoding Base64 MIME*, mempunyai konsep yang berdasarkan *RFC 1421* versi *PEM*. Sedangkan *MIME* diakhiri dengan *padding “=”* pada hasil akhir *encoding*.

3. *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*

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:

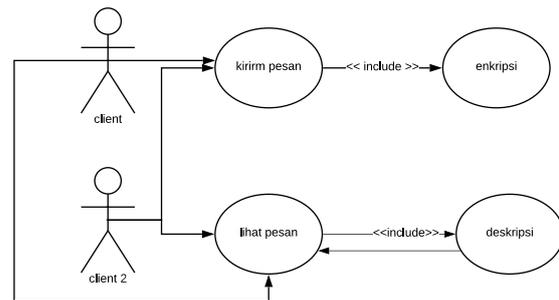
3.1 Analisis

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 CBC*

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*

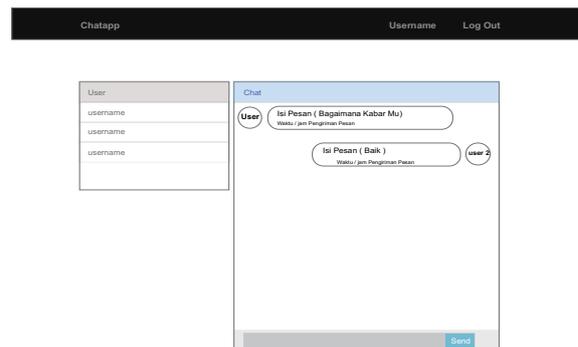
3.2 Desaign Sistem



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 inialisasi *array*
2. *liblary Django* di gunnakan sebagai *framework* yang menghubungkan 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 tampilan 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 *bootstrap* 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 enkripsi dengan menggunakan metode AES CBC dan base64 ini

dilakukan dengan menggunakan kunci dan *initial vector* yang berbeda yang digunakan 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. 1 Black Box 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. 2 Pengujian AES-CBC dan Base64

No	Panjang Karakter	enkripsi	Deskripsi
1	197	berhasil	Berhasil

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

No	Panjang karakter	enkripsi	Deskripsi
28	191	berhasil	berhasil
29	275	berhasil	Tidak berhasil
30	245	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 algoritma AES CBC dan Base64r ini, ada beberapa kesimpulan yang didapat diantaranya:

1. Aplikasi *chating* mampu mengimplementasikan algoritma *kriptografi* dengan hasil yang didapatkan dari 30 kali pengujian pada data pesan dengan hasil 17 data pesan dari keseluruhan data yang telah berhasil di enkripsi. Nilai yang didapatkan dari hasil pengujian adalah 57% pesan yang terdeskripsi yang memiliki kesamaan dengan pesan asli, menggunakan kunci dan *initial vector* yang sama untuk membuka pesan yang terenkripsi. Hasil pengujian dari total data didapat 13 pesan setelah melakukan pengujian sebanyak 30 kali pengujian didapatkan hasil 43% pesan yang terdeskripsi tidak memiliki kesamaan dengan pesan asli karena menggunakan kunci dan *initial vector* yang berbeda untuk membuka enkripsi
 - a. Data pesan yang berhasil di deskripsi diuji menggunakan kunci dan *initial vector* yang sama sedangkan pengujian yang menggunakan kunci dan *initial*

vector yang berbeda tidak berhasil di deskripsi

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.