Bulletin of Information System Research (BIOS)
Volume 1, No 3, August 2023 Page: 99-109
ISSN 2963-2455 (media online)
https://journal.grahamitra.id/index.php/bios

Implementasi Mode Operasi Cipher Block Chaining (CBC) Untuk Mengoptimalkan Algoritma Affine Cipher Dalam Pengamanan Data

Salman Siregar

Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Email: perunggu5@gmail.com

Abstrak−Kerahasiaan dan keamanan data adalah hal yang sangat penting dalam komunikasi data, baik dengan tujuan keamanan bersama, maupun untuk privasi individu. Para pengguna komputer yang menginginkan agar datanya tidak diketahui oleh pihak-pihak yang tidak berkepentingan selalu berusaha menyiasati cara mengamankan informasi yang akan dikomunikasikan atau yang akan disimpan. Perlindungan terhadap kerahasiaan data meningkat, salah satu caranya dengan menerapkan ilmu kriptografi. Kriptografi adalah salah satu ilmu yang digunakan untuk menjaga kerahasiaan dan keamanan data sudah berkembang sejak jaman Yunani kuno. Kriptografi semakin berkembang dari jaman kejaman sampai saat ini. Salah satu metode kriptografi yang cukup handal, stabil dan menjadi induk dari algoritma – algoritma kriptografi yang popular saat ini adalah Cipher Block Chaining (CBC). Cipher Block Chaining (CBC), mode ini merupakan mekanisme umpan balik (feedback) pada sebuah blok, dan dalam hal ini hasil enkripsi blok sebelumnya di umpan balikkan kedalam enkripsi blok yang current. Caranya, blok plaintext yang current di-XOR-kan terlebih dahulu dengan blok ciphertext hasil enkripsi sebelumnya, selanjutnya hasil peng-XOR-an ini masuk kedalam fungsi enkripsi. Dengan algoritma Cipher Block Chaining (CBC), setiap blok ciphertext begantung tidak hanya pada blok plaintext nya tetapi juga pada seluruh blok plaintext sebelumnya.

Kata Kunci: Kriptografi; Penyandian Teks; Algoritma Cipher Block Chaining

Abstract−Confidentiality and security of data is of paramount importance in data communication, both for the purpose of shared security, and for individual privacy. Computer users who want their data not to be known by unauthorized parties always try to find ways to secure the information to be communicated or to be stored. Protection of data confidentiality is increasing, one way is by applying the science of cryptography. Cryptography is one of the sciences used to maintain the confidentiality and security of data that has been developing since ancient Greece. Cryptography is growing from the cruel era to the present. One of the cryptographic methods that is quite reliable, stable and is the parent of the popular cryptographic algorithms today is Cipher Block Chaining (CBC). Cipher Block Chaining (CBC), this mode is a feedback mechanism on a block, and in this case the results of the previous block encryption are fed back into the current block encryption. The trick, the current plaintext block is XORed first with the previously encrypted ciphertext block, then the XOR results are entered into the encryption function. With the Cipher Block Chaining (CBC) algorithm, each ciphertext block depends not only on its plaintext block but also on all previous plaintext blocks.

Keywords: Cryptography; Text Encoding; Cipher Block Chaining Algorithm

1. PENDAHULUAN

Keamanan data pada era teknologi saat ini sangatlah diperlukan, karena pengguna aplikasi sosial media dalam berkomunikasi semakin meningkat. Data yang dikomunikasikan dapat saja berupa data penting atau rahasia Aplikasi-aplikasi sosial media seperti *facebook*, *instagram*, *twitter* dan lain-lain, sangat rentan akan pembobolan dan pencurian data. Oleh karena itu, sangat penting dilakukan pengamanan terhadap data yang didistribusikan melalui aplikasi-aplikasi tersebut. Teknik kriptografi, steganografi, *watermaking* merupakan teknik yang umum dan dapat digunakan dalam upaya pengamanan data penting atau rahasia sehingga dapat meminimalkan tindakan-tindakan pencurian data.

Kriptografi merupakan bidang pengetahuan yang menggunakan persamaan matematis untuk melakukan proses enkripsi maupun dekripsi. Teknik ini digunakan untuk mengkonversi data kedalam bentuk kode-kode tertentu sehingga menjadi susunan huruf acak yang terurut dan tidak dapat dibaca. Berdasarkan penelitian terdahulu, mengatakan bahwa kriptografi adalah suatu ilmu yang mempelajari bagaimana cara menjaga proses pendistribusian data atau pesan dari pengirim kepada penerima tanpa mengalami gangguan dari pihak ketiga. Kriptografi merupakan cabang dari ilmu matematika yang memiliki banyak fungsi dalam pengamanan data [1][2]. Kriptografi memiliki proses mengambil pesan atau *message* dan menggunakan beberapa fungsi untuk menggenerasi materi kriptografis. Teknik kriptografi melakukan proses pengamanan data berdasarkan algoritma. Salah satu algoritma kriptografi klasik yang dapat digunakan dalam mengamankan data adalah *Affine Cipher*.

Affine Cipher adalah algoritma kriptografi yang dikembangkan dari metode *caesar cipher*. Perbedaan yang mendasar dari algoritma ini adalah pergeseran dilakukan dengan cara melakukan perkalian terhadap suatu bilangan yang relatif prima dengan bilangan yang digunakan pada saat proses dekripsi. Kelemahan algoritma ini adalah mudah diserang dengan hanya melihat *ciphertext*-nya. Kesesuaian kemunculan frekuensi *ciphertext* dengan frekuensi pada *plaintext* pada umumnya dapat memudahkan pemcahan algoritma ini. Berdasarkan penelitian terdahulu, mengatakan bahwa kunci pada algoritma *affine cipher* hanya memiliki 25 kemungkinan kunci untuk alfabet dan 128 kemungkinan kunci untuk ASCII [3]. Salah satu solusi yang dapat dilakukan untuk mengatasi kelemahan algoritma *vegenere cipher* adalah menerapkan salah satu mode operasi kriptografi pada prosedur algoritma.

Cipher Block Chaining (CBC), mode ini merupakan mekanisme umpan balik (*feedback*) pada sebuah blok, dan dalam hal ini hasil enkripsi blok sebelumnya diumpan balikkan ke dalam enkripsi blok yang *current*. Caranya, blok plainteks yang *current* di-XOR-kan terlebih dahulu dengan blok *ciphertext* hasil enkripsi sebelumnya, selanjutnya hasil peng-XOR-an ini masuk ke dalam fungsi enkripsi. Berdasarkan mode CBC, setiap blok *ciphertext* begantung tidak hanya pada blok plainteksnya tetapi juga pada seluruh blok plainteks sebelumnya. Berdasarkan penelitian terdahulu, menyimpulkan bahwa proses enkripsi dan deskripsi pada mode CBC telah memenuhi kaidah yang ada dalam metode kriptografi sehingga penggunaan mode operasi ini dapat menghindari pembacaan dengan konsep blok yang monoton pada plainteks dan cipherteks [4].

Penelitian ini menguraikan bagaimana mengoptimalkan keamanan data berdasarkan algoritma *affine*. Optimalisasi pengamanan data dilakukan dengan menerapkan mode operasi CBC terhadap *cipher* yang dihasilkan berdasarkan algoritma *affine cipher*. Proses ini dapat menghasilkan *cipher* yang lebih sulit dipecahkan oleh pihak lain, sehingga keamanan data yang bersifat penting atau rahasia lebih terjamin.

2. METODOLOGI PENELITIAN

2.1 Kriptografi

Kriptografi telah dikenal dan dipakai cukup lama sejak tahun 1900 sebelum masehi pada prasasti prasasti kuburan. Kriptografi sendiri berasal dari kata *Crypto* yang berarti rahasia dan graph yang berarti tulisan. Sehingga dapat dikatakan kriptografi adalah tulisan yang tersembunyi. Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data serta otentifikasi. Sedangkan ilmu dan seni memecahkan *ciphertext* disebut kriptanalisis, pelakunya disebut kriptanalis. Cabang ilmu yang mempelajari keduanya, yaitu teknik kriptografi [5][6][7][8].

Secara umum pengertian kriptografi adalah cabang ilmu yang mempelajari cara mengubah informasi dari keadaan/bentuk normal (dapat dipahami) menjadi bentuk yang tidak dapat dipahami. Kriptografi merupakan ilmu matematika yang berhubungan dengan transformasi data untuk membuat artinya tidak dapat dipahami (untuk menyembunyikan maknanya), mencegahnya dari perubahan tanpa izin, atau mencegahnya dari penggunaan yang tidak sah. Pesan asli disebut sebagai *plaintext* dan pesan yang telah disandikan disebut *ciphertext*. Pesan terakhir yang telah disandikan dan kemudian dikirim disebut kriptogram. Proses mengubah *plaintext* menjadi *ciphertext* disebut enkripsi atau *enciphering*. Kebalikan dari proses tersebut, yaitu mengubah *ciphertext* menjadi *plaintext* disebut dekripsi atau *deciphering*. Teknik kriptografi memiliki beberapa teknik penyandian data yaitu simetris dan asimetris. Kriptografi simetris menggunakan kunci yang sama (kunci simetris) untuk melakukan proses enkripsi dan dekripsi. Kriptografi asimetris menggunakan kunci yang berbeda untuk proses enkripsi (menggunakan kunci publik) dan dekripsi (menggunakan kunci *private*) [9][10][11].

2.2 Algoritma Affine Cipher

Affine Cipher adalah teknik *cipher* yang merupakan perluasan dari *Caesar Cipher*. *Affine Cipher* tergolong dalam algoritma klasik yang merupakan algoritma penyandian yang sudah ada sebelum era digital sekarang ini [12]. Algoritma klasik pada dasarnya hanya terdiri dari *cipher* subtitusi dan *cipher* tranposisi. *Cipher* subtitusi yaitu proses mensubtitusi karakter-karakter yang ada pada *plaintext*. Sedangkan *cipher* tranposisi yaitu proses pertukaran huruf-huruf yang terdapat dalam suatu *string*.

Teknik kriptografi *Affine Cipher* merupakan salah satu tehnik dalam kriptografi klasik yang cukup sederhana sehingga sangat rentan terhadap kriptanalisis. Algoritma *Affine Cipher* adalah salah satu algoritma yang termasuk ke dalam kategori kriptografi klasik dengan jenis *cipher subsitusi*. Pada penerapannya, algoritma ini memiliki kelemahan yaitu memiliki ukuran kunci yang kecil. Berdasarkan ukuran kunci yang kecil menyebabkan algoritma ini dapat dipecahkan dengan pencarian *brute force* [13]. Kelebihan *Affine cipher* adalah mempunyai algoritma yang dapat dimodifikasi dengan berbagai teknik. Modifikasi yang dapat dilakukan pada *affine cipher* adalah menggabungkan algoritma *Affine cipher* dengan *cipher* lain, mengganti kunci *Affine cipher* dengan berbagai fungsi dan matriks dan memperluas ruang plainteks dan cipherteks pada *Affine cipher*. *Affine cipher* juga dapat diaplikasikan pada *stream cipher* sebagai pembangkit *keystream.*

Affine Cipher bukanlah *cipher* yang aman, sebab kuncinya (m dan b) dapat ditemukan dengan *exhaustive key search*, karena menggunakan alfabet yang hanya 26 huruf, maka hanya ada 25 pilihan untuk nilai b dan hanya ada 12 buah nilai m yang relatif prima dengan 26 yaitu 1, 3, 5, 7, 9, 15, 17, 19, 21, 23, dan 25, dengan mencoba semua kombinasi m dan b, maka nilai m dan b yang cocok dapat ditemukan dengan mudah. Salah satu cara memperbesar faktor kerja untuk *exhaustive key search*, enkripsi sebaiknya tidak dilakukan terhadap huruf individual, tetapi dalam blok-blok huruf [14]. Langkah-langkah enskripsi berdasarkan algoritma *affine cipher* [14][15] adalah :

1. *Affine cipher* adalah perluasan dari *Caesar cipher*, yang mengalihkan *plaintext* dengan sebuah nilai dan menambahkannya dengan sebuah pergeseran. Secara matematis enkripsi *plainkext* menghasilkan *ciphertext* dinyatakan dengan fungsi kongruen:

$$C \equiv mP + b \pmod{n}. \quad (1)$$

Bulletin of Information System Research (BIOS)
Volume 1, No 3, August 2023 Page: 99-109
ISSN 2963-2455 (media online)
https://journal.grahamitra.id/index.php/bios

2. *Affine chiper* pada metode *affine* adalah perluasan dari metode *Caesar Cipher*, yang mengalikan *plaintext* (P) dengan sebuah nilai (a) dan menambahkannya dengan sebuah pergeseran (k). P menghasilkan *ciphertext* C dinyatakan dengan fungsi kongruen:

   C=((a x P) +k) mod 26 (2)

Langkah-langkah deskripsi [16][17], yaitu :

1. Proses dekripsi *Affine Cipher* adalah dimana 26 adalah jumlah *alphabet*, persamaan 1 digunakan pada proses enkripsi. Proses dekripsi menggunakan persamaan 2 di bawah ini :

   P= a-1 (Ci-k) mod 26 (3)

   a adalah bilangan bulat yang harus relatif prima dengan 26. Dengan kata lain *great common divisior* gcd(a,26) harus sama dengan 1.
2. Sementara deskripsi *ciphertext* menjadi *plaintext* sebagai berikut:

   P ≡ m-1 (C-b)(mod n) (4)

   dimana:
   n = ukuran alfabet
   p = *plaintext* yang dikonversi menjadi bilangan bulat dari 0 sampai sesuai dengan urutan dalam alfabet
   C = *ciphertext* yang dikonversi menjadi bilangan bulat dari 0 sampai n-1 sesuai dengan urutan dalam alfabet
   m = bilangan bulat yang harus relatif prima dengan n (jika tidak relatif prima, deskripsi tidak bisa dilakukan)
   b = jumlah pergeseran

2.3 Mode Operasi *Cipher Block Chainning*

Cipher Block Chainning (CBC) dibuat dengan tujuan dapat mengimplementasikan enkripsi dan dekripsi *text*, dan dapat memperlihatkan prosedur dan hasil yang di dapatkan untuk mengamankan pesan *text* khususnya prosedur yang dilakukan pada kegiatan penyandian dan pengembalian *text* ke bentuk semula. Sebagai input yang dibutuhkan oleh system adalah *text* yang diinput langsung oleh *user* dapat diinputkan merupakan karakter-karakter ASCII 255. Berdasarkan analisa yang dilakukan terhadap data *text*, maka data *text* hanya disimpan dalam bentuk *text* saja tidak aman segingga siapa pun masih bisa membaca dan mengerti isinya. Salah satu teknik yang umum digunakan dalam mengamankan data adalah melakukan penyandian terhadap *text* dari data-data tersebut, sehingga *text* asli tersebut tidak dapat dimengerti oleh orang lain kecuali dilakukan proses pengembalian kebentuk pesan asli.

Cipher Block Chainning (CBC) merupakan penerapan mekanisme umpan balik pada sebuah blok bit dimana hasil enkripsi blok sebelumnya diumpan balikkan ke dalam proses enkripsi *block current*. Caranya, blok *plaintext* yang *current* di-XOR-kan terlebih dahulu dengan blok *ciphertext* hasil enkripsi sebelumnya, selanjutnya hasil peng-XOR- an ini masuk ke dalam fungsi enkripsi. Berdasarkan setiap blok *ciphertext* tidak hanya bergantung pada blok *plaintext*-nya tetapi juga pada seluruh blok plaintext sebelumnya. Dekrepsi dilakukan dengan memasukkan blok *ciphertext* yang *current* ke fungsi dekripsi, kemudian meng-XOR-kan hasil dengan blok *ciphertext* sebelumnya. Blok *ciphertext* sebelumnya berfungsi sebagai umpan maju (*feedforward*) pada akhirnya proses dekripsi. Secara matematis, enkripsi dan dekripsi dengan algoritma CBC dinyatakan sebagai berikut [18][19]:

Ci Ek(Pi⊕ci– 1) (5)
Pi = Dk(Ci⊕ci–1) (6)

Cipher Block Chainning bekerja dengan mode *block* yaitu melakukan pengelompokkan biner-biner *plaintext* menjadi beberapa kelompok sesuai dengan ketentuan yang ditetapkan oleh pengguna (orang yang mengenkripsikan pesan). Proses enkripsi maupun dekripsi dilakukan dengan meng XOR kan setiap nilai blok dengan blok sebelumnya kemudian hasil yang didaparkan dari operasi XOR di XOR kan kembali dengan kunci. Biner hasil operasi XOR pada setiap blok akan digeser keposisi kiri atau kanan dengan jumlah yang ditentukan okeh pengguna sistem. Nilai awal dank unci ditetapkan sebelum proses enkripsi maupun dekripsi dilakukan dan harus disepakati oleh pelaku enkripsi dan pelaku dekripsi. Panjang kunci dan nilai awal (*Initial Vector*/C0) harus sama dengan jumlah bit perkelompok, artinya apabila jumlah bit perkelompok sama dengan 8 bit, maka jumlah kunci dan C0 adalah 8 bit.

3. HASIL DAN PEMBAHASAN

Keamanan data di era digital saat ini sangat penting dan diperlukan untuk menjaga data terutama data atau informasi yang sifatnya penting atau rahasia. Kemajuan ilmu pengetahuan dan teknologi saat ini sangat berdampak luas terhadap peningkatan pengetahuan untuk melakukan berbagai tindakan pembobolan terhadap data penting atau rahasia yang didistribusikan melalui media sosial dan media lainnya yang tekoneksi dengan internet. Mengabaikan keamanan data akan sangat beresiko kepada pengguna sosial media, termasuk akan mengalami kehilangan data data yang bersifat penting atau rahasia.

Kriptografi merupakan salah satu ilmu yang yang berperan penting dalam bidang pengamanan data atau informasi. Kriptografi memiliki teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi

misalnya kerahasiaan dan integritas data, serta otentikasi. Pemanfaatan berbagai algoritma dari teknik kriptografi dapat dimanfaatkan menjadi salah satu upaya dalam meminimalkan tindakan-tindakan penyalahgunaan data penting atau rahasia yang dapati dilakukan oleh pihak-pihak yang tidak bertanggung jawab.

Kuat lemahnya algoritma kriptografi tidak terletak dari hasil enkripsi atau dekripsi yang dihasilkan dari proses pengamanan, melainkan terletak pada kerahasiaan kunci yang digunakan. Setiap metode yang ada dalam algoritma kriptografi memiliki kelemahan masing-masing yang salah satu kelemahan tersebut terletak pada kuncinya. Kunci merupakan jantung pertahanan pengamanan data dalam teknik kriptografi. Semakin bagus kunci yang digunakan, maka semakin kuat pengamanan data yang dihasilkan.

Salah satu algoritma dari teknik kriptografi yang memiliki kelemahan kunci adalah algoritma *affine cipher*. Kelemahan kunci dari algoritma ini terjadi karena algoritma ini masih merupakan perluasan dari *caesar cipher* dan tergolong dalam algoritma klasik yang merupakan algoritma penyandian yang sudah ada sebelum era digital sekarang ini, sehingga kunci dari algoritma ini masih dapati dicari menggunakan teknik *brute force*.

3.1 Penerapan Algoritma *Affine Cipher*

Teknik kriptografi a*ffine cipher* merupakan salah satu tehnik dalam kriptografi klasik yang cukup sederhana sehingga sangat rentan terhadap kriptanalisis. Algoritma *affine cipher* adalah salah satu algoritma yang termasuk ke dalam kategori kriptografi klasik dengan jenis *cipher subsitusi*. Algoritma ini memiliki kelemahan yaitu memiliki ukuran kunci yang kecil. Ukuran kunci yang kecil menyebabkan algoritma ini dapat dipecahkan dengan pencarian *brute force*. Kelebihan *affine cipher* adalah mempunyai algoritma yang dapat dimodifikasi dengan berbagai teknik.

Plaintext yang digunakan pada contoh kasus ini adalah teks yang diinput oleh pengguna, dimisalkan *plaintext* adalah SALMAN_SIREGAR yang akan disandikan menjadi *ciphertext*. *Plaintext* tersebut dirubah terlebih dahulu ke desimal dan hasilnya sebagai berikut :

Tabel 1. Nilai desimal dan biner *plaintext*

Karakter *Plaintext*	*Decimal*	Biner
S	83	01010011
A	65	01000001
L	76	01001100
M	77	01001101
A	65	01000001
N	78	01001110
_	95	01011111
S	83	01010011
I	73	01001001
R	82	01010010
E	69	01000101
G	71	01000111
A	65	01000001
R	82	01010010

1. Pemrosesan Berdasarkan Mode Operasi CBC
   Plaintext yang digunakan sebagai contoh dalam penelitian ini terlebih dahulu dienkripsi berdasarkan mode oeprasi CBC, kemudian dilakukan enkripsi berdasarkan algoritma *Affine Cipher*.
   a. Penentuan Kunci dan IV/C_0
      Proses enkripsi berdasarkan mode operasi CBC membutuhkan nilai biner vektor awal (C_0) dan kunci yang ditentukan panjang bit-nya. Contoh kasus ini menggunakan nilai C_0 dan kunci dengan panjang 16 bit, dimana Kunci = XY dan C_0 = AB
      Berikutnya kunci terlebih dahulu dan dirubah kedalam *decimal* setelah itu dirubah kedalam bentuk biner.

      $X = 88_{decimal} = 01011000_{biner}$

      $Y = 89_{decimal} = 01011001_{biner}$

      Kemudian digabungkan menjadi :

      Kunci = 0101100001011001 (16 bit)

      Berikutnya karakter *Initial Vector* atau $C_{0,}$ dirubah dalam bentuk desimal dan biner :

      $A = 65_{decimal} = 01000001_{biner}$

      $B = 66_{decimal} = 01000010_{biner}$

      Kemudian digabungkan menjadi:

      IV/C_0 = 0100000101000010 (16 bit)

Bulletin of Information System Research (BIOS)
Volume 1, No 3, August 2023 Page: 99-109
ISSN 2963-2455 (media online)
https://journal.grahamitra.id/index.php/bios

b. Kelompokkan Biner-biner *plaintext* sesuai dengan jumlah bit kunci
Tahap ini merupakan tahap pengelompokkan niilai-nilai biner *plaintext* sesuai dengan jumlah bit kunci ataupun jumlah C0. Pada kasus ini jumlah bit kunci maupun C0 adalah 16 bit, oleh karena itu, *plaintext* harus dikelompokkan menjadi 16 bit setiap kelompok (16 bit = 2 karakter).

Tabel 2. Pengelompokkan *Plaintext*

Blok	*Plaintext*	Desimal	Biner
P1	S A	83 65	01010011 01000001
P2	L M	76 77	01001100 01001101
P3	A N	65 78	01000001 01001110
P4	_ S	95 83	01011111 01010011
P5	I R	73 82	01001001 01010010
P6	E G	69 71	01000101 01000111
P7	A R	65 82	01000001 01010010

c. Proses Enkripsi Berdasarkan Mode Operasi CBC
Adapun sususan dari algoritma *Cipher Block Chaining* (CBC) dalam proses enkripsi adalah sebagai berikut :

Ci = Ek (Pi ⊕ ci – 1)
CP_1 = Blok P1⊕C_0
P1 = 01010011 01000001
C0 = 01000001 01000010⊕
= 00010010 00000011
Kunci = 01011000 01011001⊕
= **0100**1010 01011010

Geser empat bit ke kanan, sehingga menjadi 101001011010**0100** ($A5A4_{16}$)

CP_2 = Blok P2⊕C2 -1
P2 = 01001100 01001101
C2-1 = 10100101 1010**0100**⊕
= 11101001 11101001
Kunci = 01011000 01011001⊕
= **1011**0001 10110000

Geser empat bit ke kanan, sehingga menjadi 000110110000**1011** ($1B0B_{16}$)

CP_3 = Blok P3⊕C3-1
P3 = 01000001 01001110
C3-1 = 00011011 0000**1011**⊕
= 01011010 01000101
Kunci = 01011000 01011001⊕
= **0000**0010 00011100

Geser empat bit ke kanan, sehingga menjadi 0010000111000**0000** ($21C0_{16}$)

CP_4 = Blok P4⊕C4-1
P4 = 01011111 01010011
C4-1 = 00100001 1100**0000**⊕
= 01111110 10010011
Kunci = 01011000 01011001⊕
= **0010**0110 11001010

Geser empat bit ke kanan, sehingga menjadi 011011001010**0010** ($6CA2_{16}$)

CP_5 = Blok P5⊕C5-1
P5 = 01001001 01010010
C5-1 = 01101100 1010**0010**
= 00100101 11110000
Kunci = 01011000 01011001⊕
= **0111**1101 10101001

Geser empat bit ke kanan, sehingga menjadi 110110101001**0111** ($DA97_{16}$)

CP_6 = Blok P6⊕C6-1
P6 = 01000101 01000111
C6-1 = 11011010 1001**0111**⊕
= 10011111 11010000
Kunci = 01011000 01011001⊕
= **1100**0111 10001001

Geser empat bit ke kanan, sehingga menjadi 011110001001**1100** ($789C_{16}$)

CP_7 = Blok P7⊕C7-1
P7 = 01000001 01010010

Bulletin of Information System Research (BIOS)
Volume 1, No 3, August 2023 Page: 99-109
ISSN 2963-2455 (media online)
https://journal.grahamitra.id/index.php/bios

C7-1 = 01111000 1001**1100**⊕
= 00111001 11001110
Kunci = 01011000 01011001⊕
= **0110**0001 10010111

Geser empat bit ke kanan, sehingga menjadi 000110010111**0110** (1976_{16})

Hasil dekripsinya adalah **A5A41B0B21C06CA2DA97789C1976.** Berdasarkan proses di atas, maka diperoleh hasil penerapan mode operasi CBC adalah :

Tabel 3. Hasil Mode Operasi CBC

Cipher	Hexa	Desimal	Biner
C1	A5	165	10100101
C2	1B	27	00011011
C3	21	33	00100001
C4	6C	108	01101100
C5	DA	218	11011010
C6	78	120	01111000
C7	19	25	00011001
C8	A4	164	10100100
C9	0B	11	00001011
C10	C0	192	11000000
C11	A2	162	10100010
C12	97	151	10010111
C13	9C	156	10011100
C14	76	118	01110110

2. Proses Enkripsi berdasarkan Algoritma *Affine Cipher*
Hasil pemrosesan mode operasi CBC pada pesan asli akan dienkripsi kembali berdasarkan algoritma *affine cipher*. *Affine cipher* melibatkan kunci pada proses enkripsi. Proses enkripsi berdasarkan *affine cipher* adalah Ci = ((m x Pi) + b) Mod n. Nilai Pi diperoleh dari nilai-nilai Ci hasil mode operasi CBC. Bila pada contoh kasus ini nilai m= 23 , b = 7 dan nilai modulus (n) yang digunakan adalah 256.

C1 = ((m x Pi)+b) mod 256
= (23 x 165) + 7) mod 256
= 3802 mod 256
= 218 = ┌
C2 = ((m x Pi)+b) mod 256
= (23 x 27)+7) mod 256
= 628 mod 256
= 116 = t
C3 = ((m x Pi)+b) mod 256
= (23 x 33)+7 mod 256
= 766 mod 256
= 254 = ■
C4 = ((m x Pi)+b) mod 256
= (23 x 108)+7 mod 256
= 2491 mod 256
= 187 = ╗
C5 = ((m x Pi)+b) mod 256
= (23 x 218)+7 mod 256
= 5021 mod 256
= 157 = **Ø**
C6 = ((m x Pi)+b) mod 256
= (23 x 120)+7 mod 256
= 2767 mod 256
= 207 = ¤
C7 = ((m x Pi)+b) mod 256
= (23 x 25)+7 mod 256
= 582 mod 256
= 70 = F
C8 = ((m x Pi)+b) mod 256 n
= (23 x 164)+7 mod 256
= 3779 mod 256
= 195 = ├

Bulletin of Information System Research (BIOS)
Volume 1, No 3, August 2023 Page: 99-109
ISSN 2963-2455 (media online)
https://journal.grahamitra.id/index.php/bios

C9 = ((m x Pi)+b) mod 256
= (23 x 11)+7 mod 256
= 260 mod 256
= 4 = ♦
C10 = ((m x Pi)+b) mod 256
= (23x 192)+7 mod 256
= 4423 mod 256
= 71 = G
C11 = ((m x Pi)+b) mod 256
= (23 x 162)+7 mod 256
= 3733 mod 256
= 149 = u
C12 = ((m x Pi)+b) mod 256
= (23x x 151)+7 mod 256
= 3480 mod 256
= 152 = ÿ
C13 = ((m x Pi)+b) mod 256
= (23 x 156)+7 mod 256
= 3595 mod 256
= 11 = ♂
C14 = ((m x Pi)+b) mod 256
= (23 x 118)+7 mod 256
= 2721 mod 256
= 161 = í

Tabel 4. Hasil Proses Enkripsi Affine Cipher

Cipher	Desimal	Karakter
C1	218	┌
C2	116	t
C3	254	■
C4	187	╗
C5	157	Ø
C6	207	¤
C7	70	F
C8	195	├
C9	4	♦
C10	71	G
C11	149	u
C12	152	ÿ
C13	11	♂
C14	161	í

Berdasarkan proses enkripsi di atas, maka diperoleh *ciphertext* akhir adalah ┌ t■╗Ø ¤F├♦Guÿ♂í *Ciphertext* inilah yang didistribusikan kepada penerima pesan.

3. Proses Dekripsi Berdasarkan Algoritma *Affine Cipher*
Proses dekripsi berdasarkan algoritma *affine cipher* dilakukan berdasarkan formulasi Pi = ((m^{-1} x Ci) + b) Mod n. Nilai m^{-1} merupakan nilai *invers* (balikan) dari nilai m yang digunakan pada proses enkripsi. Oleh karena itu, sebelum dilakukan proses dekripsi, terlebih dahulu dicari nilai *invers* dari nilai m.
Nilai invers dari m dicari dengan menentukan nilai perkalian dari nilai m dengan syarat nilai m dikalikan dengan sebuah bilangan bulat positif dan bila hasil perkalian tesebut dimoduluskan dengan nilai n, maka nilainya ekuivalen dengan 1 mod n, sehingga :
m^{-1} = (nilai m x .....) mod n ≅ 1 mod n
bila diasumsikan bilangan positif yang dipilih adalah 167, maka :
m^{-1} = (23 x 167) mod n ≅ 1 mod 256
m^{-1} = (23 x 167) mod 256 ≅ 1
m^{-1} = 3841 mod 256 ≅ 1
m^{-1} = 1 ≅ 1, sehingga pemilihan nilai 167 memenuhi syarat dan dapat digunakan sebagai *invers* nilai m. Sehingga m^{-1} = 167.
Proses dekripsi :
P1 = (m^{-1} x (Ci-b)) mod n
= (167 x (218-7)) mod 256
= (167 x 211) mod 256

= 35237 mod 256
= 165

P2 = (m^{-1} x (Ci-b)) mod n
= (167 x (116-7)) mod 256
= (167 x 109) mod 256
= 18203 mod 256
= 27

P3 = (m^{-1} x (Ci-b)) mod n
= (167 x (254-7)) mod 256
= (167 x 247) mod 256
= 41249 mod 256
= 33

P4 = (m^{-1} x (Ci-b)) mod n
= (167 x (187-7)) mod 256
= (167 x 180) mod 256
= 30060 mod 256
= 108

P5 = (m^{-1} x (Ci-b)) mod n
= (167 x (157-7)) mod 256
= (167 x 150) mod 256
= 25050 mod 256
= 218

P6 = (m^{-1} x (Ci-b)) mod n
= (167 x (207-7)) mod 256
= (167 x 200) mod 256
= 33400 mod 256
= 120

P7 = (m^{-1} x (Ci-b)) mod n
= (167 x (70-7)) mod 256
= (167 x 63) mod 256
= 10521 mod 256
= 25

P8 = (m^{-1} x (Ci-b)) mod n
= (167 x (195-7)) mod 256
= (167 x 188) mod 256
= 31396 mod 256
= 164

P9 = (m^{-1} x (Ci-b)) mod n
= (167 x (4-7)) mod 256
= (167 x -3) mod 256
= -501 mod 256
= -245 (11)

P10 = (m^{-1} x (Ci-b)) mod n
= (167 x (71-7)) mod 256
= (167 x 64) mod 256
= 10688 mod 256
= 192

P11 = (m^{-1} x (Ci-b)) mod n
= (167 x (149-7)) mod 256
= (167 x 142) mod 256
= 23714 mod 256
= 162

P12 = (m^{-1} x (Ci-b)) mod n
= (167 x (152-7)) mod 256
= (167 x 145) mod 256
= 24215 mod 256
= 151

P13 = (m^{-1} x (Ci-b)) mod n
= (167 x (11-7)) mod 256
= (167 x 4) mod 256
= 668 mod 256
= 156

Bulletin of Information System Research (BIOS)
Volume 1, No 3, August 2023 Page: 99-109
ISSN 2963-2455 (media online)
https://journal.grahamitra.id/index.php/bios

P14 = (m^{-1} x (Ci-b)) mod n
= (167 x (161-7)) mod 256
= (167 x 154) mod 256
= 25718 mod 256
= 118

Berdasarkan proses dekripsi di atas, maka diperoleh pra-*plaintext* hasil dekripsi berdasarkan *affine cipher* adalah

Tabel 5. Hasil Dekripsi Berdasarkan *Affine Cipher*

Pra-Plain	Desimal	Hexa	Biner
P1	165	A5	10100101
P2	27	1B	00011011
P3	33	21	00100001
P4	108	6C	01101100
P5	218	DA	11011010
P6	120	78	01111000
P7	25	19	00011001
P8	164	A4	10100100
P9	11	0B	00001011
P10	192	C0	11000000
P11	162	A2	10100010
P12	151	97	10010111
P13	156	9C	10011100
P14	118	76	01110110

4. Proses Dekripsi Berdasarkan Mode Operasi CBC

Proses dekripsi berdasarkan mode oeprasi CBC dilakukan dengan menggunakan nilai kunci dan C0 yang sama dengan yang digunakan pada proses enkripsi mode CBC, dimana nilai kunci dan C0 memiliki panjang 16 bit. Karakter C0 adalah AB dan karakter kunci adalah XY.

Biner Kunci = 0101100001011001 (16 bit)
Biner C0 = 0100000101000010 (16 bit)

Selanjutnya biner-biner pra-*plaintext* yang didapatkan dari proses dekripsi berdasarkan algoritma *affine cipher* (tabel 3.5), dikelompokkan sesuai dengan jumlah bit kunci dan C0 yaitu 16 bit per kelompok.

Tabel 6. Pengelompokkan Biner Pra-*Plaintext*

Pra-Plain	Biner	
P1	10100101 00011011	A51B
P2	00100001 01101100	216C
P3	11011010 01111000	DA78
P4	00011001 10100100	19 A4
P5	00001011 11000000	0B C0
P6	10100010 10010111	A297
P7	10011100 01110110	9C76

Sebelum dilakukan proses dekripsi berdasarkan mode operasi CBC, maka terlebih dahulu dilakukan proses pengembalian 4 bit kanan dari masing-masing kelompok pra plaintext di atas ke sebelah kiri, karena pada proses enkripsi sebelumnya telah dilakukan proses right shift (pergeseran ke kanan) sebanyak 4 bit.

C1 = 101001011010**0100** Geser empat bit ke kiri : **0100**101001011010
C2 = 000110110000**1011** Geser empat bit ke kiri : **1011**000110110000
C3 = 001000011100**0000** Geser empat bit ke kiri : **0000**001000011100
C4 = 011011001010**0010** Geser empat bit ke kiri : **0010**011011001010
C5 = 110110101001**0111** Geser empat bit ke kiri : **0111**110110101001
C6 = 011110001001**1100** Geser empat bit ke kiri : **1100**011110001001
C7 = 000110010111**0110** Geser empat bit ke kiri : **0110**000110010111

Langkah terakhir adalah, melakukan proses dekripsi dengan cara yang sama seperti pada proses enkripsi.

P1 = C1⊕C0
= 01001010 01011010
= <u>01000001 01000010⊕</u>
= 00001011 00011000
= <u>01011000 01011001⊕</u>
= 01010011 01000001 = S A

P2 = C2⊕C2-1
= 10110001 10110000

Bulletin of Information System Research (BIOS)
Volume 1, No 3, August 2023 Page: 99-109
ISSN 2963-2455 (media online)
https://journal.grahamitra.id/index.php/bios

= 10100101 10100100⊕
= 00010100 00010100
= 01011000 01011001⊕
= 01001100 01001101 = L M

P3 = C3⊕C3-1
= 00000010 00011100
= 00011011 00001011⊕
= 00011001 00010111
= 01011000 01011001⊕
= 01000001 01001110 = A N

P4 = C4⊕C4-1
= 00100110 11001010
= 00100001 11000000⊕
= 00000111 00001010
= 01011000 01011001⊕
= 01011111 01010011 = _ S

P5 = C5⊕C5-1
= 01111101 10101001
= 01101100 10100010⊕
= 00010001 00001011
= 01011000 01011001⊕
= 01001001 01010010 = I R

P6 = C6⊕C6-1
= 11000111 10001001
= 11011010 10010111⊕
= 00011101 00011110
= 01011000 01011001⊕
= 01000101 01000111 = E G

P7 = C7⊕C7-1
= 01100001 10010111
= 01111000 10011100⊕
= 00011001 00001011
= 01011000 01011001⊕
= 01000001 01010010 = A R

Biner-biner yang dihasilkan dari proses dekripsi di atas, dekelompokkan menjadi 8 bit setiap kelompok untuk mendapatkan karakter dari plaintext akhir.

Tabel 7. Hasil Dekripsi Berdasarkan mode Operasi CBC

Desimal	Biner	Karakter Plaintext
01010011	83	S
01000001	65	A
01001100	76	L
01001101	77	M
01000001	65	A
01001110	78	N
01011111	95	_
01010011	83	S
01001001	73	I
01010010	82	R
01000101	69	E
01000111	71	G
01000001	65	A
01010010	82	R

Sehingga diperoleh hasil deskripsi adalah **SALMAN_SIREGAR** yang merupakan karakter yang sama seperti *plaintext* awal.

4. KESIMPULAN

Berdasarkan analisa dan pengujian yang telah dilakukan dalam penelitian ini, maka disimpulkan beberapa hal, sebagai berikut Pengamanan data rahasia yang dilakukan berdasarkan algoritma kriptografi dapat dioptimalkan dengan mengkombinasikannya dengan mode operasi yang ada di dalam teknik kriptografi, salah satunya mode

operasi *Cipher Block Chainning*. *Ciphertext* yang dihasilkan dari proses kombinasi CBC dan *affine cipher* sangat acak, karena terjadi sebelum dienkripsi berdasarkan algoritma *affine cipher*, maka terlebih dahulu dilakukan proses pengacakan bit data, sehingga dapat meningkatkan keamanan data rahasia. Aplikasi keamanan data rahasia yang dibangun dapat bekerja sesuai dengan algoritma yang digunakan dalam penelitian ini, sehingga sangat membantu dan mempermudah pengguna untuk mengamankan data rahasia yang didistribusikan kepada orang lain.

REFERENCES

[1] R. Fadillah, A. S. Idris, D. M. lumban Gaol, G. Lubis, R. Meisisri, and M. Syahrizal, "Implementasi Algoritma Fast Encryption Algorithm (FEAL) Dan Algoritma Fibonacci Mengamankan File Teks," in *Prosiding Seminar Nasional Sosial, Humaniora, dan Teknologi*, 2022, pp. 295–300.
[2] Wibowo, S., Nilawati, F. E., & Suharnawi. (2014). Implementasi Enkripsi Dekripsi Algoritma Affine Cipher Berbasis Android. Techno.COM, Vol. 13, No. 4, 13(4), 215–221.
[3] Belakang, A. L. (2010). BAB I PENDAHULUAN A. Latar Belakang. 1–8.
[4] Karim, A. (2017). Penerapan Algoritma Cipher Block Chaining (Cbc) Dan Wrapping Bit Untuk Pengamanan File Citra Digital. March, 392–400.
[5] Muhammad Fairuzabadi, "Implemetasi Kriptografi Klasik menggunakan Borland Delphi," Jurnal Dinamika Informatika, pp. 65-78, September 2010. (2010). September, 2010.
[6] http://bkpsdmd.babelprov.go.id/content/keamanan-data-informasi.
[7] Sadiki, R, 2012. Kriptografi Untuk Keamanan Jaringan, Edisi I, Andi, Yogyakarta. Septiarini, A dan Hamdani, 2011. Sistem Kriptografi untuk Text Message Menggunakan Metode Affine, Jurnal Informatika Mulawarman, Vol 6 No. 1
[8] Praptomom, Y. (2005). KEAMANAN SISTEM INFORMASI Yuli Praptomo PHS STMIK El Rahma – Yogyakarta. STMIK El Rahma – Yogyakarta.
[9] A. Menezes, Oorschot, P. van, & Vanstone, S. (2011). Handbook of Applied Cryptography.
[10] Zulfikar, iqbal.M, 2019, Kriptografi Untuk Keamanan Pengiriman Email Menggunakan Blowfish dan Rivest Shamir Adleman (RSA), SNATI, ISSN : 1907-5022.
[11] Oppliger, Rolf. 2005. Contemporary Cryptography. London: Artech House, Inc.
[12] Religia, Y., Studi, P., Informatika, T., Komputer, F. I., Dian, U., & Semarang, N. (n.d.). Implementasi Algoritma Affine Cipher Dan Vigenere Cipher Untuk Keamanan Login.
[13] Sebayang, & Sari, A. M. (2015). Implementasi Kombinasi Beaufort Cipher Dan Affine Cipher Pada Three-Pass Protocol Untuk Pengamanan Data. Http://repository.usu.ac.id. http://repository.usu.ac.id/handle/123456789/44871
[14] R. Munir, "Kriptografi," Inform. Bandung, 2006.
[15] Rachmawati, D., & Candra, A. (2015). Implementasi Kombinasi Caesar dan Affine Cipher untuk Keamanan Data Teks. Jurnal Edukasi Dan Penelitian Informatika (JEPIN), 1(2).
[16] Rauf, ruzlan akba. Kode ascii lengkap. http://informatikakbaruzlan.blogspot.com/2013/05/kodeascii-lengkap.html. Tanggal akses 14 Desember 2013.
[17] Ari sugandi. (2020). Mode operasi block cipher. Id.wikipedia.org. https://id.wikipedia.org/wiki/Mode_operasi_block_cipher.
[18] Dewi Rosmala (2012), Implementasi Mode Cipher Block Chaining (CBC) pada pengamanan Data, Vol.3, 2012.
[19] Andriani, D. (2017). Perancangan Aplikasi Penyandian Teks Dengan Menggunakan Algoritma Chiper Block Chaining. Jurnal Teknik Informatika Unika St. Thomas (JTIUST), 02(338), 14–23.