Jurnal Sains Komputer & Informatika (J-SAKTI)
Volume (2) No.1 Maret 2018, pp. 47-56
ISSN:2548-9771/EISSN:2549-7200
http://tunasbangsa.ac.id/ejurnal/index.php/jsakti

Penyandian *File Word* Berdasarkan Algoritma *Rivest Code 5* (RC5)

Widodo Arif Prabowo[1], Annisa Fitri Harahap[2], Ridha Ismadiah[3]
[1,2,3]Mahasiswa Prorgram Studi Teknik Informatika STMIK Budidarma Medan
[1,2,3]Jln. Sisingamangaraja No. 338 Sp. Limun Medan
widodoarif795@gmail.com

Abstract

Nowadays, important data in the form of word files have been widely used. But there are still few who apply security techniques to the important files. Files that are confidential or important, if they fall into the hands of others may be misused or manipulated for certain purposes. The existence of document security applications built on cryptographic algorithms is one solution to solve the above problems. Cryptographic techniques secure a data or important files by encoding the data into a cipher that is difficult to understand again by others. The RC5 *algorithm is one of the cryptographic technique algorithms that can be used to encode text from word files based on the* RC5 *algorithm so that it can improve the expression of important and confidential word files.*

Keywords: *Cryptography, Stream Cipher,* RC5*, File, Word*

Abstrak

Saat ini, data penting yang berbentuk file word telah banyak digunakan. Namun masih sedikit yang menerapkan teknik keamanan pada file penting tersebut. File yang bersifat rahasia atau penting, bila jatuh ke tangan pihak lain tentu dapat disalahgunakan atau dimanipulasi pada tujuan-tujuan tertentu. Adanya aplikasi pengamanan dokumen yang dibangun berdasarkan algoritma kriptografi merupakan salah satu solusi untuk menyelesaikan masalah di atas. Teknik kriptografi mengamankan sebuah data atau file penting dengan menyandikan data tersebut menjadi sandi yang susah dipahami lagi oleh orang lain. Algoritma RC5 *merupakan salah satu dari algoritma teknik kriptografi yang dapat digunakan untuk menyandikan teks dari file word berdasarkan algoritma* RC5*, sehigga dapat meningkatkan pengaanan dari file word yang sifatnya penting dan rahasia.*

Kata Kunci*: Kriptografi, Cipher Aliran,* RC5*, File, Word*

1. PENDAHULUAN

Pertukaran data melalui jaringan komputer terutama *internet*, sangat mungkin dilakukan karena tentunya akan mempercepat dan memudahkan proses pertukaran data terutama untuk pertukaran data dengan jarak yang jauh. Sebagai contoh adalah pertukaran data yang dilakukan oleh sebuah kantor pusat yang ditujukan kepada kantor cabang.

Salah satu jenis data yang sering dipergunakan untuk pertukaran data melalui media *internet* diantaranya seperti file *word* dan lain sebagainya. Selain kemudahan yang diperoleh, ada juga bahaya yang muncul dalam proses pertukaran data melalui media *internet*, salah satunya adalah pencurian data yang dilakukan pihak ketiga yang tidak bertanggungjawab yang bertujuan untuk kepentingan pribadi. Berdasarkan penelitian sebelumnya, mengatakan bahwa media *internet* sekarang menjadi jalur pertukaran yang sangat vital untuk isi file

yang bersifat rahasia [1]. Oleh karena itu, dibutuhkan teknik pengamanan data untuk menghindari penyadapan terhadap konten data yang digunakan sebagai objek pertukaran.

Kriptografi merupakan salah satu teknik yang dapat digunakan dalam mengamankan data yang bersifat rahasia atau pribadi. Proses transformasi informasi yang berlangsung dua arah yang terdiri dari proses enkripsi dan dekripsi adalah ruanglingkup dari kriptografi[2][3]. Berdasarkan penelitian sebelumnya yang dilakukan oleh Setyaningsih mengatakan bahwa teknik kriptografi sangat penting diimplementasikan untuk melindungi data yang ditransmisikan melalui suatu jaringan komunikasi[4].

Metode RC5 merupakan salah satu dari algoritma kriptografi primitif yang merupakan sasaran pengkajian RC5. Data *dependent rotations* merupakan suatu teknik yang dapat merotasi data secara sirkuler sebanyak N rotasi. Algoritma RC5 menggunakan metode simetrik dan pengolahan dalam bentuk blok *chiper*. Jumlah putaran pada algoritma RC5 disimbolkan dengan r yang memiki nilai antara 1, 2, 3, 4, ..., 225, jumlah kata dalam bit di simbolkan dengan w. Jumlah yang didukung adalah 16 bit, 32 bit, dan 64 bit, kata kunci (*key word*) disimbolkan dengan b dengan range 1, 2, 3, 4,..., 225 [1]. Ada 3 proses utama dalam RC5, yaitu perluasan kunci, enkripsi dan dekripsi. Perluasan kunci merupakan proses membangkitkan kunci internal dengan memanfaatkan komputasi rotasi *left* regular *shift* (<<<) dan *right* regular *shift* (>>>), dengan panjang kunci tergantung dari jumlah putaran[5].

Penelitian menguraikan bagaimana mengamankan *file word* berdasarkan algoritma RC5 dengan membangun sebuah aplikasi yang mampu mengenkripsi dan dekripsi teks dari *file word*. Hal ini dilakukan agar isi *file word* tidak dapat dipahami maknanya oleh orang lain sehingga dapat meminimalkan tindakan-tindakan penyalahgunaan, manipulasi atau pemanfaatan *file* tersebut pada hal-hal yang merugikan pemiliki *file*.

2. METODOLOGI PENELITIAN

2.1 File Word

File adalah entitas dari data yang disimpan di dalam sistem *file* yang dapat diakses dan diatur oleh pengguna. Sebuah *file* memiliki nama yang unik dalam direktori dimana ia berada. Alamat direktori dimana suatu berkas ditempatkan diistilahkan dengan *path*. Salah satu contoh *file* adalah *file word*. *File word* merupakan jenis *file* yang dihasilkan oleh aplikasi *microsoft office* yang umumnya digunakan sebagai aplikasi pengolah kata dalam bentuk surat, dokumen. Dokumen yang dibuat dan dikelolah pada aplikasi *microsoft office* disimpan dalam format *file word* [7].

2.2 Kriptografi

Berdasarkan terminologinya, kriptografi dapat didefenisikan sebagai ilmu dan seni untuk menjaga keamanan pesan ketika informasi didistribusikan dari suatu tempat ke tempat lain [7]. Kriptografi pada awalnya merupakan ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya[8]. Secara umum, penerapan algritma kriptografi dalam pengamanan data rahasia harus mencapai beberapa

aspek yaitu kerahasiaan (*confidentiality*), integritas data (*data integrity*), otetikasi (*authentication*) dan ketiadaan penyangkalan (*non repudiation*)[9].

Kriptografi memerlukan algoritma untuk mewujudkan tujuannya sebagai salah satu teknik keamanan data. Keamanan data berdasarkan teknik kriptografi tidak dijamin oleh algritmanya, melainkan ditentukan oleh kunci yang digunakan pada saat proses enkripsi dan dekripsi dilakukan. Hal inilah yang menyebabkan peranan kunci dalam penerapan algoritma kriptografi sangat penting.

2.3 Algoritma RC5

Rivest Code-5 (RC5) merupakan salah satu algoritma *cipher simetri* yang bekerja dengan sisyem *block cipher*. RC5 dirancang oleh Profesor Ron Rivest dari Laboratorium RSA MIT. RC5 di publikasikan pada Desember 1994. Algoritma ini dirancang sedemikian rupa sehingga memenuhi syarat-syarat berikut [5][10] :

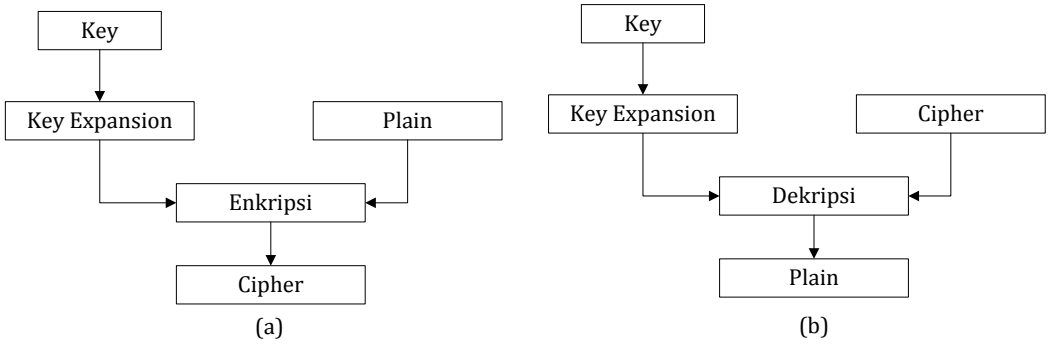
a. RC5 harus dirancang menjadi algoritma *cipher simetri.*
b. RC5 harus cocok untuk digunakan pada *hardware* dan *software.*
c. RC5 harus berkecepatan tinggi.
d. RC5 harus dapat beradaptasi pada berbagai panjang *word*. Contohnya, pada *prosesor* terbaru 64-bit, panjang *word*-nya lebih panjang daripada prosesor 32-bit. RC5 harus dapat memanfaatkan ini, oleh karena itu RC5 memiliki parameter w yang menandakan panjang *word*.
e. RC5 harus dapat beroperasi dalam berbagai jumlah ronde. Jumlah ronde yang bervariasi memungkinkan pengguna untuk memanipulasi RC5 untuk menjadi lebih cepat atau lebih aman.
f. RC5 harus dapat beroperasi dalam berbagai panjang kunci. Hal ini mengakibatkan panjang kunci b menjadi parameter dalam algoritma RC5.
g. RC5 harus berstruktur sederhana. Struktur yang sederhana belum tentu menghasilkan keamanan yang rendah. Struktur yang sederhana akan memungkinkan analisis dan evaluasi yang cepat untuk menentukan kekuatan algoritma RC5.
h. RC5 harus hemat dalam pemakaian memori. Hal ini akan memungkinkan implementasi RC5 ke dalam *smart-card* atau perangkat lain yang memiliki keterbatasan memori.
i. RC5 harus mengimplementasikan metode data-*dependent rotations*. Metode ini adalah primitif kriptografi yang merupakan sasaran pengkajian RC5. Data *dependent rotations* adalah suatu teknik yang merotasi data yang sekarang diproses secara sirkuler sebanyak N, dimana besarnya N tergantung data yang lain.

3. HASIL DAN PEMBAHASAN

3.1 Analisa Masalah

Masalah keamanan *file word* memang menjadi hal yang penting untuk diperhatikan, mengingat saat ini penggunaan terhadap *file* jenis *word* untuk mengarsipkan informasi rahasia ini semakin banyak. Penyandian berdasarkan algoritma dari teknik kriptografi memberikan hasil yang signifikan untuk meminimalisir tindakan-tindakan penyalahgunaan file tersebut. Pengamanan file word berdasarkan algoritma RC5 meliputi dua tahap utama, yaitu proses

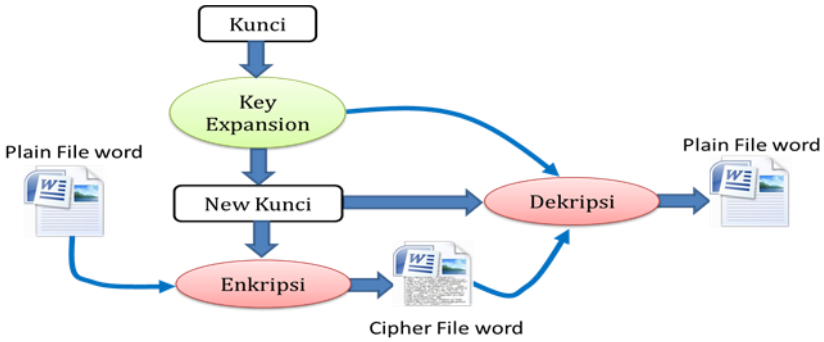
perluasan kunci (*key expansion*), proses enkripsi atau dekripsi. Berikut ini disajikan diagram alur kerja algoritma RC5 secara umum :



Gambar 1. Skema Enkripsi dan Dekripsi

Berdasarkan gambar 1 di atas, terlihat bahwa proses utama yang harus dilakukan baik pada saat melakukan enkripsi maupun dekripsi adalah proses *key expansion*, kemudian dilanjutkan dengan proses enkripsi maupun dekripsi.

Plaintext yang menjadi *input* dalam penelitian ini adalah teks dari *file word* yang akan diamankan. Teks *file word* tersebut dibagi menjadi dari dua *word* w-bit, yang ditandai dengan A dan B. *RC5* menggunakan *expanded key table* (*key table* yang diperluas), S[0…t – 1], terdiri dari t = 2(r + 1) *word* w-bit. Algoritma *key expansion* menginisialisasi S dari parameter key rahasia dari yang diberikan oleh *user*. Tabel S dalam *enkripsi* RC5 bukan dan tidak sama dengan tabel S-Box seperti yang yang digunakan pada DES. RC5 menggunakan *entry* dalam S secara sekuensial, satu pada satu waktu. Skema penerapan algoritma RC5 pada penyandian teks *file word* ditunjukkan pada gambar di bawah ini .



Gambar 2. Skema Proses Enkripsi dan Dekripsi File Word

3.2 Implementasi

Proses implementasi berikut ini akan mencontohkan secara manual pemanfaatan algoritma RC5 untuk menyandikan teks *file word*. Diasumsikan isi teks dari *file word* yang akan disandikan adalah **WIDODO** dan kunci yang digunakan adalah **252567**.

a. Proses Ekspansi Kunci (*Key Expansion*)
   Langkah awal adalah membuat inisialisasi panjang kunci 6 bit. Kunci: 252567 diinisialisasikan menjadi :

K0 = [2]
K1 = [5]
K2 = [2]
K3 = [5]
K4 = [6]
K5 = [7]

Kemudian melakukan proses pencarian nilai *Array* S yang terakhir dengan melakukan perulangan dengan cara menjumlahkan inisialisasi S-Box dengan inisialisasi kunci kemudian hasil keduanya di *mod* dengan jumlah *bit plaintext* yang di *inputkan* serta menginisialisasikan i dan j adalah 0 agar dapat menghitung *pseudo-code random* (r).

untuk i = 0, maka
j = (j + S [i] + k [i] mod 6
= (j + S [0] + k [0] mod 6)
= (0 + 0 + 2) mod 6
j = 2
Swap S [0], S [1], sehingga menghasilkan :
Array S0=[2], S1=[1], S2=[0], S3=[3], S4=[4], S5=[5]

untuk i = 1 maka,
j = (j + S [i] + k [i] mod 6)
= (j + S [1] + k [1] mod 6)
= (2 + 1 + 5) mod 6
j = 2
Swap S[1], S[0], sehingga menghasilkan :
Array S0=[1], S1=[2], S2=[0], S3=[3], S4=[4], S5=[5]

untuk i = 2 maka,
j = (j + S [i] + k [i] mod 6
= (j + S [2] + k [2] mod 6
= (2 + 2 + 2) mod 6
j = 0
Swap S[2], S[3], sehingga menghasilkan:
Array S0=[1], S1=[2], S2=[0], S3=[3], S4=[4], S5=[5]

untuk i = 3 maka :
j = (j + S [i] + k [i] mod 6
= (j + S [3] + k [3] mod 6
= (0 + 3 + 5) mod 6
j = 2
Swap S[3], S[4] menghasilkan :
Array S0=[1], S1=[3], S2=[0], S3=[2] S4=[4], S5=[5]

untuk i = 4 maka :

j = (j + S [i] + k [i] mod 6
= (j + S [4] + k [4] mod 6
= (2 + 4 + 6) mod 6
j = 0
Swap S[4], S[5] menghasilkan:
Array S0=[1], S1=[3], S2=[4], S3=[2] S4=[0], S5=[5]

untuk i = 5 maka :
j = (j + S [i] + k [i] mod 6
= (j + S [5] + k [5] mod 6
= (0 + 5 + 7) mod 6
j = 0
Swap S[5], S[0] menghasilkan :
Array S0=[1], S1=[3], S2=[4], S3=[2] S4=[5], S5=[0]

Setelah mendapatkan hasil *array* S dari langkah keempat, maka proses selanjutnya yaitu meng-*XOR* kan *plaintext* sebanyak 4 kali dikarenakan *plaintext* yang akan dienkripsi berjumlah 4 karakter. Hal ini menyebabkan dibutuhkannya 1 kunci dan 1 kali pengoperasian *XOR* untuk tiap-tiap karakter pada *plaintext*. *Array* yang digunakan untuk meng-*XOR* kan adalah *array* dari hasil pencarian nilai *array* terakhir. *array* S terakhir adalah array S0=[1], S1=[3], S2=[4], S3=[2], S4=[5], S5=[0]. Berikut proses pembentukan kunci :

untuk Kunci K[0] :
i = (0 + 1) mod 6
= 1
j = (0 + S [1]) mod 6
= (0 + 3) mod 6
= 3
Swap (S[1], S[3])
Array S = S0=[1], S1=[3], S2[4], S3=[2], S4=[5], S5=[0]
K0 = S [(S[1] + S[3]) mod 6]
= S [((3 + 2) mod 6)]
= S [(5 mod 6)]
= S[5]
= 0
K[0] = 00110000

untuk Kunci K[1] :
i = (1 + 1) mod 6
= 2
j = (1 + S [2] mod 6)
= (3+ 4) mod 6
= 1
Swap (S[2], S[1])
Array S = S0=[0], S1=[3], S2[4], S3=[2], S4=[5], S5=[1]
K1 = S [(S [2] + S [1] mod 6)]

= S [((4 + 3) mod 6)]
= S [(7 mod 6)]
= S [1]
= 3
K[1] = 00110011

untuk Kunci K[2] :
i = (2 + 1) mod 6
I = 3
j = (1+ S [3] mod 6)
= (1+2) mod 6)
= 4 mod 6
j = 3
Swap S[3], S[3]
Array S = S0=[0], S1=[3], S2[4], S3=[2], S4=[5], S5=[1]
K2 = S [(S[3] + S[3]) mod 6]
= S [(2 + 2) mod 6]
= S [4]
= 5
K[2] = 00110101

untuk Kunci K[3] :
i = (3 + 1) mod 6
i = 4
j = (3 + S[4]) mod 6
j = (3 + 5) mod 6
j = 2
Swap S[4], S[2]
Array S = S0=[0], S1=[3], S2[5], S3=[2], S4=[4], S5=[1].
K3 = S [(S [4] + S [2]) mod 6]
= S [(4 + 5) mod 6]
= S [9 mod 6]
= S [3]
= 2
K[3] = 00110010

untuk Kunci K[4] :
i = (4 + 1) mod 6
i = 5
j = (2 + S[5]) mod 6
j = (2 + 1) mod 6
j = 3
Swap S[5], S[3]
Array S = S0=[0], S1=[3], S2[5], S3=[2], S4=[4], S5=[1].
K4 = S [(S [5] + S [3]) mod 6]
= S [(1 + 2) mod 6]

= S [3 mod 6]
= S [3]
= 2
K[4] = 00110010

untuk Kunci K[5] :
i = (5 + 1) mod 6
i = 0
j = (3 + S[0]) mod 6
j = (3 + 0) mod 6
j = 3
Swap S[0], S[3]
Array S = S0=[0], S1=[3], S2[5], S3=[4], S4=[2], S5=[1].
K5 = S [(S [0] + S [3]) mod 6]
= S [(0 + 4) mod 6]
= S [4 mod 6]
= S [4]
= 2
K[5] = 00110010

Berdasarkan proses *key expassion*, maka telah didapatkan kunci untuk proses enkripsi maupun dekripsi adalah :

Tabel 1. Hasil Key Expansion

Index Kunci	Biner Kunci
K[0]	00110000
K[1]	00110011
K[2]	00110101
K[3]	00110010
K[4]	00110010
K[5]	00110010

b. Proses Enkripsi
Proses enkripsi diawali dengan mengkonversi *plaintext* ke *biner*, karena *biner* dari masing-masing *plaintext* inilah nantinya yang akan di XOR dengan masing-masing kunci.

Nilai desimal setiap karakter *plaintext* berdasarkan tabel ASCII adalah :
W = 01010111; **I** = 01001001; **D** = 01000100; **O** = 01001111
D = 01000100; **O** = 01001111

Kemudian nilai *Array* S (kunci tabel 1) di *XOR* dengan nilai biner *plaintext* yang telah dihitung dengan *key* yang ditentukan oleh *user*.

Plaintext [0]	= **W**	*Plaintext [1]*	= **I**
Biner *Plaintext*	= 01010111	*Biner Plaintext*	= 01001001
Key [0]	= 00110000 ⊕	*Key [1]*	= 00110011 ⊕
Biner Chipertext	= 01100111	*Biner Ciphertext*	= 01111010
Chipertext [0]	= **g**	*Ciphertext [1]*	**= z**
Plaintext [2]	= **D**	*Plaintext [3]*	= **O**
Biner Plaintext	= 01000100	*Biner Cipher*	= 01001111
Key [2]	= 00110101 ⊕	*Key [3]*	= 00110010
Biner Chipertext	= 01110001	*Biner Cipher*	= 01111101
Chipertext [2]	= **q**	*Ciphertext [3]*	= **}**
Plaintext [4]	= **D**	*Plaintext [5]*	= **O**
Biner Plaintext	= 01000100	*Biner Cipher*	= 01001111
Key [4]	= 00110010 ⊕	*Key [5]*	= 00110010 ⊕
Biner Chipertext	= 01110110	*Biner Cipher*	= 01111101
Chipertext [4]	= **v**	*Ciphertext [5]*	= **}**

Sehingga dipeorleh *chipertext* dari teks *file word* adalah **gzq}v}**

c. Proses Dekripsi
Proses dekipsi hampir sama dengan proses yang dilakukan pada dekripsi. Dekripsi diawali dengan *key expasion* (perluasan kunci) yang sama dengan kunci yang digunakan pada proses enkripsi, karena kunci algoritma ini bersifat simetris (kunci yang sama). Sehingga kunci yang dihasilkan pada proses dekripsi sama dengan kunci enkripsi. Proses XOR kunci dengan *array cipher* juga dilakukan dengan cara yang sama seperti pada prose dekripsi, sehingga diperoleh *plaintext* dari *file word* semula adalah **WIDODO**.

4. KESIMPULAN

Berdasarkan uraian analisa dan pembahasan, maka disimpulkan beberapa hal sebagai berikut :

a. Proses penyandian isi file *word* dapat dilakukan dengan algoritma *rivest code* 5 sehingga file *word* tersimpan di dalam media penyimpanan maupun file *word* yang dikirim dan diterima tidak bisa dibaca dan dipahami pihak lain kecuali penerima yang sah.
b. Kunci enkripsi dan dekripsi yang dihasilkan pada proses *key expansion* cukup acak dan sulit untuk ditemukan pola hubungan antara kunci awal, sehingga dapat mempersulit para penyerang untuk menemukan kunci yang sebenarnya.

DAFTAR PUSTAKA

[1] S.H. Suryawan, Hamdani, "Pengamanan Data File Dengan Menggunakan Algoritma Enkripsi Rivest Code 5 ," *Jurnal Informatika Mulawarman* , vol.8, no. 2, pp. 44-49, 2013.

[2] E. Setyaningsih, *Kriptografi & Implementasinya Menggunakan Matlab*, Yogyakarta:Andi, 2015.
[3] T. Zebua and E. Ndruru, "PENGAMANAN CITRA DIGITAL BERDASARKAN MODIFIKASI ALGORITMA RC4," *J. Teknol. Infomasi dan Ilmu Komput.*, vol. 4, no. 4, pp. 275–282, 2017.
[4] E. Setyaningsih, "Penyandian Citra Menggunakan Metode Playfair Cipher", J. Teknol., vol. 2, no. 2, pp. 213–219, 2009.
[5] Hamdani, S.H. Suryawan, A. Septiarini,"Pengujian Algoritma Rivest Code 5 Untuk Enkripsi Struktur File Dokumen," *Jurnal Informatika Mulawarman*, vol. Vol. 8 No. 2, pp. 44-49, Juni 2013.
[6] E. Aribowo, "Aplikasi Pengamanan Dokumen Office Dengan Algoritma Kriptografi Kunci Asimetris Elgamal," *J. Inform.*, vol. 2, no. 2, pp. 209–219, 2008.
[7] R. Sadikin, 2012, Kriptografi Untuk Keamanan Jaringan dan Implementasinya dalam Bahasa Java, Yogyakarta, Andi.
[8] H. Pandiangan, S. Sijabat "PERANCANGAN MEDIA PENGIRIMAN PESAN TEKS DENGAN PENYANDIAN PESAN MENGGUNAKAN ALGORITMA RC4 BERBASIS WEB ," *Jurnal Matik Penusa* , vol. Volume XIX, No. 1 , no. ISSN 2088-3943 , pp. 63-71, Juni 2016
[9] T. Zebua, "ANALISA DAN IMPLEMENTASI ALGORITMA TRIANGLE CHAIN PADA PENYANDIAN RECORD DATABASE," *Pelita Inform. Budi Darma*, vol. 3, no. 2, pp. 37–49, 2013.
[10] N. Widyastuti, "Pengembangan Metode Beaufort Cipher Menggunakan Pembangkit Kunci Chaos," *J. Teknol.*, vol. 7, no. 1, pp. 73–82, 2014.