Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)

journal homepage: http://journal.lembagakita.org/index.php/jtik

Model Pengembangan Keamanan Data dengan Algoritma *ROT 13 Extended Vernam Cipher* dan *Stream Cipher*

Yanuar Chris Milian [1*], Wiwin Sulistyo [2]

[1,2] *Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana, Kota Salatiga, Provinsi Jawa Tengah, Indonesia.*

a r t i c l e i n f o

Article history:
Received 16 July 2022
Received in revised form
25 November 2022
Accepted 16 February 2023
Available *online* April 2023

DOI:
https://doi.org/10.35870/jtik.v7i2.716

Keywords:
Data Security; ROT13; Vernam Cipher; Streams Cipher.

Kata Kunci:
Keamanan Data; ROT13; Vernam Cipher; Stream Cipher.

a b s t r a c t

This research focuses on the importance of data security in maintaining confidentiality, particularly for sensitive and critical information that should only be accessed by certain parties. Cryptography is a mathematical method used to secure data. Several methods are available, including ROT13, Vernam Cipher, and Stream Cipher. This study combines these three methods to achieve a higher level of data security. The analysis shows that the combination of these algorithms results in a stronger and more complex encryption process than using a single method. The process involves encrypting plaintext using ROT13, followed by encrypting the result using Vernam Cipher with an inputted key, and then encrypting the output again using Stream Cipher. The advantages of this combination model include enhanced data security, as demonstrated by the three-stage encryption and decryption process.

a b s t r a k

Penelitian ini berfokus pada pentingnya keamanan data dalam menjaga kerahasiaan, terutama untuk informasi sensitif dan kritis yang hanya boleh diakses oleh pihak tertentu. Kriptografi adalah metode matematika yang digunakan untuk mengamankan data. Beberapa metode yang tersedia, termasuk ROT13, Vernam Cipher, dan Stream Cipher. Penelitian ini menggabungkan ketiga metode ini untuk mencapai tingkat keamanan data yang lebih tinggi. Analisis menunjukkan bahwa kombinasi algoritma ini menghasilkan proses enkripsi yang lebih kuat dan kompleks daripada hanya menggunakan satu metode. Prosesnya melibatkan enkripsi plaintext menggunakan ROT13, diikuti dengan mengenkripsi hasilnya menggunakan Vernam Cipher dengan kunci yang dimasukkan, dan kemudian mengenkripsi output lagi menggunakan Stream Cipher. Keuntungan dari model kombinasi ini termasuk peningkatan keamanan data, seperti yang ditunjukkan oleh proses enkripsi dan dekripsi tiga tahap.

Corresponding Author. Email: yanuarmilian@gmail.com [1].

1. Latar Belakang

Permasalahan dalam pengamanan data masih menjadi sebuah aspek terpenting dalam penyimpanan data agar terhindar dari kebocoran serta penyalahgunaan oleh pihak yang tidak bertanggung jawab. Untuk itu harus ada keamanan yang berfungsi sebagai pelindung sebuah pesan. Terdapat cara untuk melakukan pengamanan data dengan teknik penyamaran yang disebut dengan kriptografi. Kriptografi adalah suatu seni dan ilmu untuk mengamankan data yang berupa pesan yang dapat dibaca (*plaintext*) menjadi pesan yang tidak dapat dibaca (*ciphertext*), sehingga yang dapat mengetahui, menghapus dan mengganti pesan tersebut hanya penerima dan pengirim pesan saja [1]. Banyak sekali algoritma atau metode yang dapat kita gunakan untuk proses keamanan data contohnya dengan menggunakan metode ROT13, *Vernam Cipher*, *Stream Cipher* dan lain sebagainya. ROT13 (Rotate 13) merupakan suptitution enkripsi cipher yang biasa diperlukan sebagai sistem informasi UNIX yang banyak terdapat pada forum online yang bermanfaat untuk perlindungan isi artikel [2]. Sistem pergeseran dari metode ROT13 dengan menggeser maju sebanyak 13 karakter berdasarkan tabel ASCII. Untuk deskripsi ROT13 dengan menggeser sebanyak 13 karakter.

Metode *Vernam Cipher* merupakan algoritma kriptografi berjenis *symmetric key*, *Vernam Cipher* merupakan salah satu algoritma block cipher tercepat diantara yg lain. Proses enkripsi *Vernam Cipher* meggunakan mekanisme operasi XOR dengan menggabungkan antara bit *plaintext* dan *keystream*. Keunggulan dari algoritma *Vernam Cipher* dibanding cipher yang lain yaitu menggunakan *pseudorando-key* yang sama panjang dengan fungsi XOR namun dari segi kelemahan algoritma *Vernam Cipher* ini yaitu hasil enkripsi yang masih Terlihat oleh mata manusia, sehingga mudah dikenali sebagai data yang telah menjalakan proses enkripsi [3]. Sedangkan *Steam cipher* adalah algoritma kriptografi yang mengekspresikan plainteks menjadi chiperteks secara individual atau satu karakter (1 byte) [4]. *Steam cipher* dalam memiliki proses ekripsi dan deskripsi lebih cepat dibandingkan block cipher dan *Stream Cipher* yang memiliki keuntungan panjang plainteks tak terbatas tetapi masih memiliki kelemahan pada pengacakan KSA (Key Scheduling Algorithm) [5].

Penelitian terdahulu dengan judul "Analisis Penerapan Modifikasi Algoritma Vigenere Cipher, Caesar Cipher, *Vernam Cipher* dan Hill Cipher Untuk Penyisipan Pesan Dalam Gambar" peneliti melakukan pengamanan dengan menyisipkan pesan sebuah gambar menggunakan algoritma LSB (least significant bit) pada pesan yang sudah terenkripsi kemumdian memodifikasi dengan menggabungkan algoritma vigenere cipher, Vernam Chiper dan Hill Chiper. Dari penelitian tersebut menghasilkan suatu sistem dimana pesan dapat disembunyikan di dalam gambar dengan meletakkan pesan didalam bit terakhir gambar [6]. Pratama dan Dimas Aditya dengan penelitiannya tentang aplikasi keamanan teks SMS menggunakan metode *Stream Cipher*, ROT13 dan Caesar Cipher berbasis android dapat ditarik kesimpulan bahwa dengan adanya kegunaan kriptografi dalam pesan teks SMS pada android maka pencuri data tidak dapat mengambil informasi dari SMS. Serta dari penelitian tersebut menghasilkan sebuah aplikasi keamanan teks SMS yang diterapkan pada perangkat android [7]

Pada penelitian sebelumnya yang berjudul aplikasi enkripsi dan deskripsi dengan teknik XOR menggunakan metode Vernam Chiper diperoleh kesimpulan dengan aplikasi enkripsi dan deskripsi yang dikerjakan dengan teknik XOR menggunakan algoritma Vernam Chiper sangat mempermuda untuk mengirimkan pesan rahasia/senditif. Konsistensi pesan tetap terlindungi dengan keamanan pesan yang tidak dapat dibaca ataupun dimengerti oleh pihak yang tidak bertanggung jawab [8]. Dari penelitian terdahulu dengan judul 'Penerapan Kriptografi Algoritma Blowfish pada Pengamanan Pesan Data Teks". Hasil implementasi algoritma Blowfish menggunakan MS Visual data berhasil dienkripsi maupun didekripsi dan dapat kembali seperti semula, sehingga dapat digunakan untuk melakukan pengamanan data. Hasil pengujian waktu eksekusi menunjukkan proses enkripsi membutuhkan waktu yang lebih lama daripada proses dekripsi. Data yang diperoleh menunjukkan proses deksripsi 33% lebih cepat daripada proses enkripsi [9].

Tujuan mengkombinasi 3 metode tersebut untuk memperkuat algoritma *ROT13* yang bekerja hanya dengan melakukan pergeseran karakter saja. Untuk itu akan dikombinasikan dengan *Vernam Cipher* dan *Stream Cipher* yang mempunyai performa yang lebih cepat serta tak terbatasnya plainteks pada *Stream*

Cipher. Dalam penelitian ini, peneliti menggunakan algoritma ROT 13 *Extended Vernam Cipher* sebagai algoritma kunci simetris untuk mengenkripsi pesan. Algoritma ini merupakan pengembangan dari algoritma *Vernam Cipher* yang memiliki keamanan yang lebih tinggi. Selain itu, peneliti juga menggunakan *Stream Cipher* sebagai algoritma kunci asimetris untuk memperkuat keamanan pesan. *Stream Cipher* memiliki kecepatan yang tinggi dalam mengenkripsi pesan dan juga memiliki keamanan yang baik. Diharapkan penelitian ini memberikan kontribusi dalam pengembangan model keamanan data yang lebih aman dan efektif yang dapat digunakan dalam pengiriman pesan rahasia yang memerlukan keamanan yang tinggi.

2. Metode Penelitian

Kriptografi

Kriptografi (*Chryptograhy*) berasal dari bahasa yunani yang terdiri dari 2 suku kata yakni Cryptos yang berarti "secret" artinya rahasia dan Graphia berarti "wtiting" yang artinya tulisan. Kriptografi merupakan ilmu atau seni yang menjaga keamanan pesan supaya pesan dari pengirim dapat tersampaikan dengan aman kepada penerima pesan [10]. Kriptografi menggunakan algoritma (cipher) dan kunci (key). Cipher adalah fungsi matematika yang dapat mengenkripsi dan mendeskripsi data dan sedangkan kunci adalah sederetan bit yang yang dibutuhkan untuk mengenkripsi dan mendeskripsi data. Pada dasarnya terdapat beberapa komponen kriptografi seperti enkripsi, deskripsi, kunci (key), *ciphertext*, *plaintext*, pesan, dan cryptanalysis [11].

ROT13

ROT13 merupakan sebuah fungsi yang memakai kode kaisar dengan melakukan pergeseran k=13. ROT13 dirancang untuk keamanan pada sistem operasi UNIX yang banyak terdapat pada forum-forum online, yang bermanfaat untuk melindungi isi artikel. Proses enkripsi dari ROT13 dengan menggeser maju karakter sebanyak 13 kali terhitung mulai 1 karakter yang ada didepannya, sedangkan untuk deskripsinya dengan menggeser mundur karakter 13 kali yang terhitung 1 karakter dibelakangnya. Pergeseran karakter tersebut dengan berdasar pada urutan karakter pada tabel ASCII. Formula yang digunakan sebagai berikut [12] :

$$Ci = (Pi + 13) \quad (1)$$

$$\mathrm{Ci} = (\mathrm{Pi} - 13) \quad (2)$$

Keterangan :
Ci = *ciphertext*
Pi = *Plaintext*

Stream Cipher

Stream Cipher merupakan algoritma kriptografi yang berjenis symetric key digunakan untuk melakukan ekripsi dan deskrispi. Untuk meghasilkan *ciphertext* digunakan matematik XOR dan XNOR sehingga aplikasi dapat melakukan proses enkripsi dan deskripsi. Algoritma *Stream Cipher* tidak dibatasi *plaintext*-nya jadi cocok untuk mengenkripsi data atau pesan yang berkelanjutan. Rumus dari metode kriptografi *Stream Cipher* sebagai berikut [13] :

$$Ci = (Pi + Ki) mod\ 256 \quad (3)$$

$$Ci = (Pi - Ki) mod\ 256 \quad (4)$$

Keterangan :
Ci = *Ciphertext*
Pi = *Plaintext*
Ki = Key
Mod = Modulus

Vernam Cipher

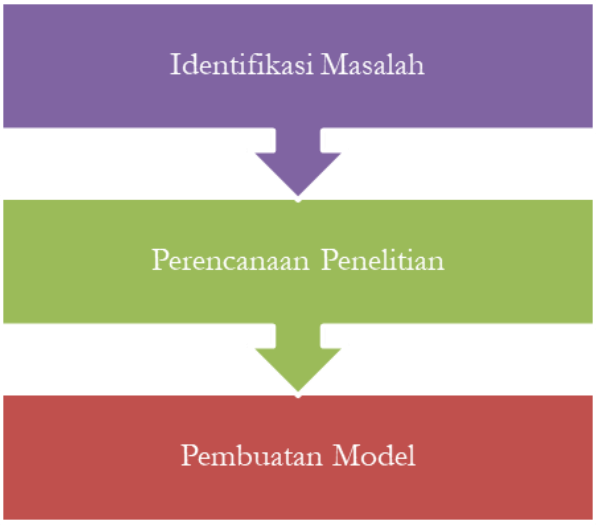
Vernam Cipher merupakan algoritma dengan sistem keamanan yang sangat baik. *Vernam Cipher* adalah metode *Stream Cipher* simetris yang mengkombinasikan *plaintext* dengan key stream dengan panjang yang sama untuk mendapatkan hasil *ciphertext* dengan memfungsikan boolean eksklusif (Xor dan Xnor). Untuk menghasilkan *ciphertext* dilakukan dengan penjumlahan modulo 2 satu bit *plaintext* dan satu bit key. Rumus matematik yang digunakan dalam metode *Vernam Cipher* sebagai berikut :

$$\mathrm{Ci} = \mathrm{Pi\ XOR\ Ki} \quad (5)$$

$$\mathrm{Pi} = \mathrm{Ci\ XOR\ Ki} \quad (6)$$

Keterangan :
Ci = Cipertext
Pi = *Plaintext*
Ki = Key

Dalam operator logika XOR, hasil akan bernilai benar jika salah satu operand bernilai benar. Dengan kata lain jika diaplikasikan dalam bit maka operator XOR akan menghasilkan 1 jika dan hanya jika salah satu operand bernilai [14]. Ada beberapa langkah yang dilakukan dalam penelitian ini untuk mencapai tujuan penelitian, dimana langkah-langkah tersebut dapat dilihat pada Gambar 1.

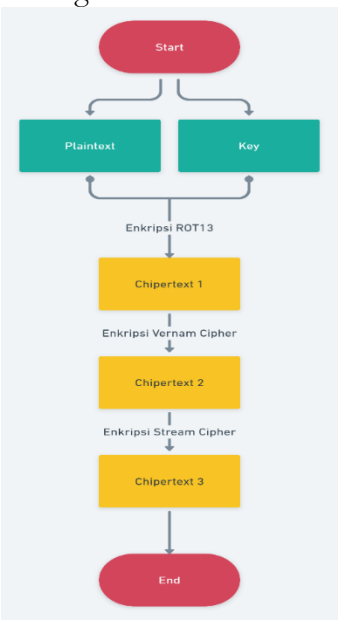


Gambar 1. Metode Penelitian

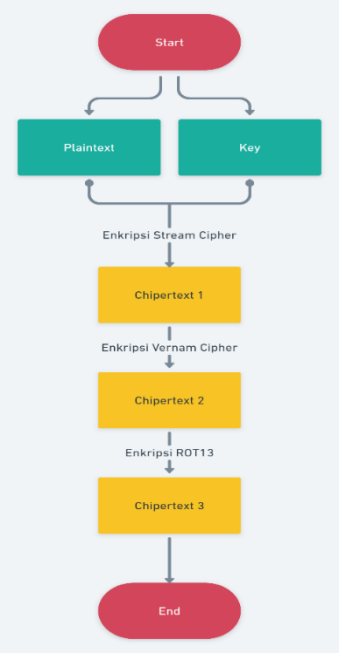
Penelitian ini dimulai dengan tahapan identifikasi masalah untuk mengetahui kekurangan atau masalah yang ada dalam dunia kriptografi. Setelah itu, dilakukan perencanaan penelitian agar masing-masing proses dalam penelitian dapat dirancang dengan baik dan sesuai dengan yang dibutuhkan. Tahapan ketiga adalah pelaksanaan penelitian atau perancangan algoritma, dimana peneliti merancang ketiga algoritma yang akan digunakan dalam penelitian ini, yaitu ROT13 Extended Vernam Cipher dan Stream Cipher. Ketiga algoritma tersebut kemudian dikombinasikan untuk menciptakan model pengembangan keamanan data. Selanjutnya, tahapan keempat adalah pengujian dan analisis hasil, di mana dilakukan pengujian terhadap kombinasi ketiga algoritma yang digunakan untuk melihat sejauh mana keamanan data dapat terjaga dengan baik. Pada tahap terakhir, yaitu tahapan kelima, dilakukan penulisan laporan penelitian yang menjelaskan hasil dari penelitian ini dan model pengembangan keamanan data yang berhasil diciptakan.

3. Hasil dan Pembahasan

Pada penelitian ini metode pengembangan keamanan data adalah dengan mengkombinasikan tiga metode yaitu metode ROT13, *Vernam Cipher* dan *Stream Cipher*. Proses kombinasi tiga metode tersebut dengan proses enkripsi dan deskripsi yang dilakukan beberapa tahap. Proses model pengembangan dapat dilihat pada gambar 2 dan gambar 3.



Gambar 2. Model Pengembangan Enkripsi



Gambar 3. Model Pengembangan Dekripsi

Dapat dilihat pada gambar 2 merupakan model pengembangan enkripsi dan gambar 3 merupakan model proses pengembangan dekripsi yang digunakan pada penelitian ini. Adapun penjelasan dari kedua model tersebut sebagai berikut :

1) Tahap pertama pada model enkripsi, pengguna memasukkan *plaintext* dan key yang ingin digunakan.
2) Tahap kedua, *plaintext* yang sudah dimasukkan akan dienkripsi menggunakan algoritma ROT13.
3) Tahap ketiga, *ciphertext* 1 hasil dari enkripsi menggunakan ROT13, dienkripsi kembali dengan key yang sudah dimasukkan diawal. Enkripsi pada tahap ini menggunakan algoritma *Vernam Cipher*.
4) Tahap keempat, *ciphertext* 2 hasil dari enkripsi di tahap ketiga, dienkripsi kembali menggunakan algoritma *Stream Cipher* dengan key yang sama. Hasil dari enkripsi ini adalah *ciphertext* 3 yang merupakan hasil akhir dari model pengembangan enkripsi.
5) Model pengembangan dekripsi merupakan kebalikan dari model pengembangan enkripsi, yaitu *ciphertext* 3 menjadi *plaintext* untuk masukkan awal, sedangkan *key*-nya tetap. Tahap ini *plaintext* dan *key* akan didekripsi dengan algoritma *Stream Cipher*.
6) Setelah itu hasil dari dekripsi awal model pengembangan dekripsi yaitu *ciphertext* 1, didekripsi kembali dengan menggunakan algoritma *Vernam Cipher*.
7) Hasil dari tahapan sebelumnya didekripsi kembali dengan ROT13. Hasil dari dekripsi adalah *plaintext* awal pada model pengembangan enkripsi atau *plaintext* asli sebelum dienkripsi

Proses Metode ROT13

Metode ROT13 adalah algoritme enkripsi sederhana yang menggunakan sandi abjad tunggal dengan pergeseran k=13.

```
29          System.out.println("==== ROT13 ====");
30          int key = 13;
31          String enkripText = "";
32          for (int a = 0; a < plaintext.length(); a++)
33          {
34              int temp = (int)plaintext.charAt(a) + key;
35              enkripText += (char)temp;
36          }
37          System.out.println("Encrypt Text : " + enkripText);
38
39          String dekripText = "";
40          String plaintext1 = enkripText;
41          for (int b = 0; b < plaintext1.length(); b++)
42          {
43              int temp = (int)plaintext1.charAt(b) - key;
44              dekripText += (char)temp;
45          }
46          System.out.println("Decrypt Text : " + dekripText);
```

Gambar 4. Algoritma ROT13

Pada Gambar 4 dapat kita lihat algoritma enkripsi-dekripsi ROT13. Sesuai dengan arti dari ROT13 pada baris 34 dan 43 koding, dapat kita lihat *plaintext* dan *plaintext*1 diubah ke dalam bentuk decimal tiap huruf, kemudian masing-masing ditambahkan dan dikurangi dengan key. Pada algoritma ini key-nya adalah 13. Contoh kerja algoritma ROT13 dengan *plaintext* ”AYO” dapat dilihat pada Tabel 1 dan Tabel 2.

Tabel 1. Konversi *Plaintext* ke Desimal dan hasil dekripsi ROT13

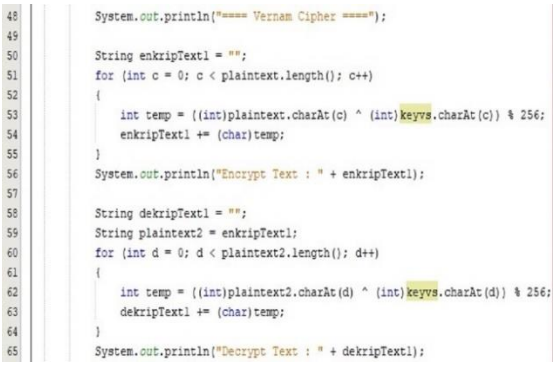
Plaintext	Desimal (ASCII)
A	65
Y	89
O	92

Tabel 2. Hasil enkripsi ROT13

Plaintext	Desimal (ASCII)
N	78
f	102
\	92

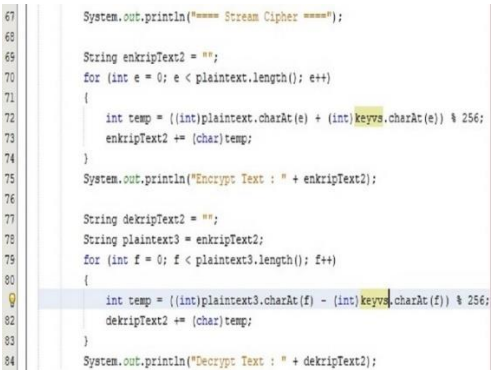
Proses Metode Vernam Cipher

Metode *Vernam Cipher* adalah algoritma berjenis *symmetric key* kunci yang digunakan untuk melakukan enkripsi dan dekripsi yang menggunakan kunci yang sama.



```
48      System.out.println("==== Vernam Cipher ====");
49
50      String enkripText1 = "";
51      for (int c = 0; c < plaintext.length(); c++)
52      {
53          int temp = ((int)plaintext.charAt(c) ^ (int)keyvs.charAt(c)) % 256;
54          enkripText1 += (char)temp;
55      }
56      System.out.println("Encrypt Text : " + enkripText1);
57
58      String dekripText1 = "";
59      String plaintext2 = enkripText1;
60      for (int d = 0; d < plaintext2.length(); d++)
61      {
62          int temp = ((int)plaintext2.charAt(d) ^ (int)keyvs.charAt(d)) % 256;
63          dekripText1 += (char)temp;
64      }
65      System.out.println("Decrypt Text : " + dekripText1);
```

Gambar 5. Algoritma *Vernam Cipher*



```
67      System.out.println("==== Stream Cipher ====");
68
69      String enkripText2 = "";
70      for (int e = 0; e < plaintext.length(); e++)
71      {
72          int temp = ((int)plaintext.charAt(e) + (int)keyvs.charAt(e)) % 256;
73          enkripText2 += (char)temp;
74      }
75      System.out.println("Encrypt Text : " + enkripText2);
76
77      String dekripText2 = "";
78      String plaintext3 = enkripText2;
79      for (int f = 0; f < plaintext3.length(); f++)
80      {
81          int temp = ((int)plaintext3.charAt(f) - (int)keyvs.charAt(f)) % 256;
82          dekripText2 += (char)temp;
83      }
84      System.out.println("Decrypt Text : " + dekripText2);
```

Gambar 6. Algoritma *Stream Cipher*

Pada Gambar 5 dapat kita lihat algoritma enkripsi-dekripsi *Vernam Cipher*. Pada baris 53 dan 62 koding, dapat kita lihat *plaintext* dan *plaintext*2 masing-masing di-XOR-kan dengan key yang sudah dimasukkan bersamaan dengan *plaintext* awal dan di-mod-kan 256, sesuai dengan rumus algoritma ini yaitu (P XOR K) mod 256. Contoh kerja algoritma *Vernam Cipher* dengan *plaintext* "AYO" dan key "123", dapat dilihat pada Tabel 3 dan Tabel 4.

Tabel 3. Konversi *Plaintext* ke Desimal dan hasil dekripsi *Vernam Cipher*

Plaintext	Desimal (ASCII)
A	65
Y	89
O	92

Tabel 4. Hasil enkripsi *Vernam Cipher*

Plaintext	Desimal (ASCII)
p	112
k	107
|	124

Proses Metode Stream Cipher

Metode *Stream Cipher* adalah algoritma sandi yang mengenkripsi data persatuan data, seperti bit, byte, nible atau per lima bit (saat data yang di enkripsi berupa data *Boudout*). Setiap mengenkripsi satu satuan data di gunakan kunci yang merupakan hasil pembangkitan dari kunci sebelumnya.

Pada Gambar 6 dapat kita lihat algoritma enkripsi-dekripsi *Stream Cipher*. Pada baris 72 dan 81 koding, dapat kita lihat *plaintext* dan *plaintext3* masing-masing ditambahkan dan dikurangi dengan *key* yang sudah dimasukkan bersamaan dengan *plaintext* awal dan di-mod-kan 256, sesuai dengan rumus algoritma ini yaitu enkripsi (P + K) mod 256 dan dekripsi (P – K) mod 256. Contoh kerja algoritma *Stream Cipher* dengan *plaintext* "AYO" dan *key* "123", dapat dilihat pada Tabel 5 dan Tabel 6.

Tabel 5. Konversi *Plaintext* ke Desimal dan hasil dekripsi *Stream Cipher*

Plaintext	Desimal (ASCII)
A	65
Y	89
O	92

Tabel 6. Hasil enkripsi *Stream Cipher*

Plaintext	Desimal (ASCII)
r	114
‹	139
‚	130

Proses Kombinasi Tiga Metode

Proses keamanan data dengan mengkombinasikan metode ROT13, *Vernam Cipher* dan *Stream Cipher* dengan mengenkripsikan pesan yang telah dibuat dan dengan kunci yang sudah dibuat sebagai bentuk langkah penyandian pesan. Tahap pertama dengan menggunakan metode ROT13 pesan dienkripsi, selanjutnya dengan menggunakan metode *Vernam Cipher* pesan tersebut dienkripsi kembali dan tahap terakhir dengan menggunakan metode *Stream Cipher* pesan dienkripsikan kembali dengan satu kali proses

pengenkripsian dan satu kunci (Key). Gambar 7 merupakan proses enkripsi kombinasi dari algoritma yang digunakan.

```
102         System.out.println("==== KOMBINASI ====");
103         int key = 13;
104         String enkripText = "";
105         String enkripText2 = "";
106         String enkripText3 = "";
107         String inttochar = "";
108
109         //KOMBINASI
110         for (int a = 0; a < plaintext.length(); a++)
111         {
112             int temp = (int)plaintext.charAt(a) + key;
113             enkripText += (char)temp;
114             temp = ((int)enkripText.charAt(a) ^ (int)keyvs1.charAt(a)) % 256;
115             enkripText2 += (char)temp;
116             temp = ((int)enkripText2.charAt(a) + (int)keyvs1.charAt(a)) % 256;
117             enkripText3 += (char)temp;
118             //inttochar += temp;
119         }
120
121         System.out.println("Encrypt Text : " + enkripText3);
```

Gambar 7. Algoritma Kombinasi Enkripsi

Contoh kerja algoritma kombinasi ini dengan *plaintext* "Teknologi" dan *key* "193264", pertama-tama *plaintext* dan *key* diubah ke dalam bentuk desimal tiap hurufnya. Baris 112-113 merupakan baris algoritma untuk *ROT13*, hasil enkripsi dapat dilihat pada tabel 9. Baris 114-115 merupakan baris algoritma *Vernam Cipher* yang mengenkripsi hasil dari algoritma *ROT13*, hasil enkripsi dapat dilihat pada tabel 10. Kemudian baris 116-117 merupakan baris algoritma *Stream Cipher* yang mengenkripsi hasil dari algoritma *Vernam Cipher*, kemudian hasil enkripsi tersebut ditampilkan pada baris 121.

Tabel 7. Konversi *plaintext* ke desimal

Plaintext	Desimal (ASCII)
T	84
e	101
k	107
n	110
o	111
l	108
o	111
g	103
i	105

Tabel 8. Konversi *key* ke desimal

Plaintext	Desimal (ASCII)
1	49
9	57
3	51
2	50
6	54
4	52

Tabel 9. Hasil enkripsi algoritma kombinasi *ROT13*

Desimal *Plaintext*	Desimal (*ROT13*)
84	97
101	114
107	120
110	123
111	124
108	121
111	124
103	116
105	118

Tabel 10. Hasil enkripsi algoritma kombinasi *Vernam Cipher*

Desimal (*ROT13*)	Desimal (*Vernam Cipher*)
97	80
114	75
120	75
123	73
124	74
121	77
124	77
116	77
118	69

Tabel 11. Hasil enkripsi algoritma kombinasi *Stream Cipher*

Desimal (*Vernam Cipher*)	Desimal (*Stream Cipher*)
80	129
75	132
75	126
73	123
74	128
77	129
77	126
77	134
69	120

Tabel 12. Hasil final enkripsi konversi ke *plaintext*

Desimal (*Stream Cipher*)	*Plaintext*
97	ü
114	ä
120	~
123	{
124	Ç
121	ü
124	~
116	â
118	w

```
129             for (int a = 0; a < plaintext.length(); a++)
130             {
131                 int temp = ((int)enkripText3.charAt(a) - (int)keyvs1.charAt(a)) % 256;
132                 dekripText += (char)temp;
133                 temp = ((int)dekripText.charAt(a) ^ (int)keyvs1.charAt(a)) % 256;
134                 dekripText2 += (char)temp;
135                 temp = (int)dekripText2.charAt(a) - key;
136                 dekripText3 += (char)temp;
137             }
138             System.out.println("Decrypt Text : " + dekripText3);
```

Gambar 8. Algoritma Kombinasi Dekripsi

Baris 131-132 merupakan baris algoritma untuk *Stream Cipher*. Baris 133-134 merupakan baris algoritma *Vernam Cipher* yang mengdekripsi hasil dari algoritma *Stream Cipher*. Kemudian baris 135-136 merupakan baris algoritma ROT13 yang mengdekripsi hasil dari algoritma *Vernam Cipher*, kemudian hasil dekripsi tersebut ditampilkan pada baris 138. Proses kerja dekripsi algoritma kombinasi ini dapat kita lihat dari kebalikan proses enkripsi pada tabel-tabel hasil enkripsi, dimulai dari *Stream Cipher*, *Vernam Cipher* dan ROT13.

Tabel 13. Hasil final dekripsi

Desimal Enkripsi	*Ciphertext*	Desimal Dekripsi	Hasil
97	ü	84	T
114	ä	101	e
120	~	107	k
123	{	110	n
124	Ç	111	o
121	ü	108	l
124	~	111	o
116	â	103	g
118	w	105	i

Hasil analisis dari kombinasi model algoritma yaitu, *plaintext* dan *key* yang sudah diinputkan akan di enkripsi *plaintext*-nya terlebih dahulu dengan menggunakan algoritma ROT13, setelah itu hasil enkripsi tersebut dienkripsi lagi menggunakan *Vernam Cipher* dengan *key* yang sudah diinputkan tadi, hasil dari enkripsi kedua ini digunakan untuk dienkripsi lagi menggunakan algoritma *Stream Cipher* dan hasil ini menjadi hasil final enkripsi. Adapun proses dekripsinya dimulai dari hasil final enkripsi, didekripsikan menggunakan *Stream Cipher* dengan *key* yang sama pada proses enkripsi, hasilnya didekripsikan dengan *Vernam Cipher* dan hasil dari dekripsi tersebut didekipsikan lagi dengan ROT13, maka akan menghasilkan *plaintext* awal. Peranan penting dalam hal ini adalah *key* yang tidak boleh dilupakan agar proses enkripsi dekripsi ini mendapatkan hasil yang sesuai.

4. Kesimpulan

Berdasarkan hasil analisis yang telah dilakukan dapat ditarik kesimpulan bahwa ketiga algoritma yang digunakan yaitu ROT13, *Vernam Cipher* dan *Stream Cipher*, dikombinasikan terlebih dahulu, cara kerjanya yaitu *plaintext* dienkripsi menggunakan ROT13, setelah itu hasil enkripsinya digunakan pada algoritma *Vernam Cipher* dan dienkripsi menggunakan key yang sudah diinputkan, hasil dari enkripsi tersebut digunakan lagi untuk dienkripsi menggunakan algoritma *Stream Cipher*, kemudian hasil enkripsi tersebut digunakan sebagai hasil akhir enkripsi. Kelebihan penggunaan model kombinasi dari ketiga algoritma ini adalah menghasilkan tingkat keamanan data menjadi lebih kuat dan sulit untuk dipecahkan, terbukti dari proses yang harus melewati tiga tahap enkripsi dan dekripsi.

5. Daftar Pustaka

[1] Azmi, F. and Erika, W., 2017. Analisis keamanan data pada block cipher algoritma Kriptografi RSA. *CESS (Journal of Computer Engineering, System and Science)*, *2*(1), pp.27-29. DOI: https://doi.org/10.24114/cess.v2i1.4967.

[2] Manullang, A.S., Puspasari, R. and Verina, W., 2020. Penyandian Database Menggunakan Metode Base64 Dan Rot13. *Jurnal Mahasiswa Fakultas Teknik dan Ilmu Komputer*, *1*(1), pp.283-292.

[3] Siahaan, K.N. and Mesran, M., 2020. Penerapan Algoritma Venigmare Cipher dan Vernam Cipher Dalam Pengamanan Data Teks. *Jurnal Sistem Komputer dan Informatika (JSON)*, *2*(1), pp.48-52. DOI: 10.30865/json.v2i1.2457.

[4] Haji, W.H. and Mulyono, S., 2012. Implementasi Rc4 Stream Cipher Untuk Keamanan Basis Data. In *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*.

[5] Lestari, D. and Riyanto, M.Z., 2012. Suatu Algoritma Kriptografi Stream Cipher Berdasarkan Fungsi Chaos. *no. November*, pp.978-979.

[6] Fatonah, S., Yulandari, A. and Ariyus, D., 2019. Analisis Penerapan Modifikasi Algoritma Vigenere Cipher, Caesar Cipher, Vernam Cipher dan Hill Cipher Untuk Penyisipan Pesan Dalam Gambar. *Jurnal VOI (Voice Of Informatics)*, *8*(2).

[7] Pratama, D.A. and Kurniawan, H., 2020. Aplikasi Keamanan Teks SMS Menggunakan Metode Stream Cipher, ROT13, Dan Caesar Cipher Berbasis Android. *Jurnal Mahasiswa Fakultas Teknik dan Ilmu Komputer*, *1*(1), pp.274-282.

[8] Firmansyah, M.H.D., 2021. Aplikasi Enkripsi Dan Dekripsi Dengan Teknik XOR Menggunakan Metode Vernam Cipher. *Kumpulan Karya Ilmiah Mahasiswa Fakultas sains dan Tekhnologi*, *2*(2), pp.8-8.

[9] Prasetiyo, B., Muslim, M.A. and Susanto, H., 2017. Penerapan Kriptografi Algoritma Blowfish pada Pengamanan Pesan Data Teks. *Techno. Com*, *16*(4), pp.358-366. DOI: 10.33633/tc.v16i4.1452.

[10] Amin, M.M., 2016. Implementasi Kriptografi Klasik Pada Komunikasi Berbasis Teks. *Pseudocode*, *3*(2), pp.129-136. DOI: 10.33369/pseudocode.3.2.129-136.

[11] Ariyus, D., 2008. *Pengantar ilmu kriptografi: teori analisis & implementasi*. Penerbit Andi.

[12] Hendrik, H., 2020. Kombinasi Algoritma Huffman dan Algoritma ROT 13 Dalam Pengamanan File Docx. *Journal of Information System Research (JOSH)*, *2*(1), pp.40-46.

[13] Manurung, D.P., Puspasari, R. and Verina, W., 2020. Perbandingan Metode Stream Dengan Metode Caesar Cipher Terhadap Pengiriman Pesan Pada Jaringan Wireless LAN. *Jurnal Mahasiswa Fakultas Teknik dan Ilmu Komputer*, *1*(1), pp.332-342.

[14] Sihombing, M.L., 2019. Penerapan Algoritma Vernam Cipher (One Time) untuk Pengamanan Login. *Pelita Informatika: Informasi dan Informatika*, *7*(3), pp.429-432.