Available at https://jtsiskom.undip.ac.id (23 May 2020)
DOI:10.14710/jtsiskom.2020.13476

Jurnal Teknologi dan Sistem Komputer, 8(3), 2020, 201-209

Implementasi vigenere cipher 128 dan rotasi bujursangkar untuk pengamanan pesan teks

Implementation of vigenere cipher 128 and square rotation in securing text messages

Rihartanto[*)], Riris Kurnia Ningsih, Achmad Fanany Onnilita Gaffar, Didi Susilo Budi Utomo

Jurusan Teknologi Informasi, Politeknik Negeri Samarinda
Jl. Cipto Mangunkusumo, Kampus Gn. Lipan, Samarinda Seberang, Samarinda, Indonesia 75131

Cara sitasi: R. Rihartanto, R. K. Ningsih, A. F. O. Gaffar dan D. S. B. Utomo, "Implementasi vigenere cipher 128 dan rotasi bujursangkar untuk pengamanan teks," *Jurnal Teknologi dan Sistem Komputer*, vol. 8, no. 3, pp. 201-209, 2020. doi: 10.14710/jtsiskom.2020.13476, [Online].

Abstract – Information that can be in the form of text, image, audio, and video, is a valuable asset that needs to be secured from unauthorized parties. This research aims to study the implementation of Vigenere cipher 128 (VC-128) and square rotation to secure text information. The square rotation is applied to increase the security of the encryption results obtained from VC-128. The randomness of the rotation results was measured using Shannon entropy based on the distance between characters, and the Avalanche Effect measured changes in the encryption results compared to the original text. The square rotation can increase the randomness of the VC-128 encryption results, as indicated by an increase in entropy values. The highest increase in entropy of 34.8 % occurs in repetitive texts with the square size that produces optimal entropy was a 9x9 medium-size square. The Avalanche effect for each test data shows inconsistent results ranging from 44.5 % to 49 %.

Keywords *– encryption optimization; square rotation; Vigenere cipher; Shannon entropy;Avalanche effect*

Abstrak - Informasi merupakan aset berharga yang keamanannya perlu dilindungi dari pihak-pihak yang tidak berhak. Menurut bentuknya, informasi dapat berbentuk teks, citra, audio dan video. Penelitian ini bertujuan untuk melindungi informasi yang disimpan dalam bentuk teks. Metode yang digunakan adalah Vigenere cipher 128 (VC-128) dan rotasi bujursangkar. Rotasi bujursangkar digunakan untuk meningkatkan keamanan dari hasil enkripsi yang diperoleh dari VC-128. Keacakan hasil rotasi diukur menggunakan entropi Shannon berdasarkan jarak antar karakter, sedangkan efek Avalanche digunakan untuk mengukur perubahan hasil enkripsi dibandingkan dengan teks aslinya. Hasil penelitian menunjukkan bahwa rotasi bujursangkar mampu meningkatkan keacakan hasil enkripsi VC-128 yang ditunjukkan dengan adanya peningkatan nilai entropi. Peningkatan entropi tertinggi sebesar 34,8 % terjadi pada teks berulang dengan ukuran bujursangkar yang menghasilkan entropi optimal adalah bujursangkar berukuran sedang, yaitu 9×9. Nilai efek Avalanche untuk setiap data uji memberikan hasil yang tidak konsisten, berkisar antara 44,5 % hingga 49 %.

Kata kunci *– optimasi enkripsi; rotasi bujursangkar; Vigenere cipher; entropi Shannon; efek Avalanche*

I. PENDAHULUAN

Informasi merupakan komoditas penting bagi siapa saja, baik bagi organisasi pemerintah, organisasi swasta, perguruan tinggi, LSM maupun individu. Pesatnya perkembangan teknologi menjadikan informasi menjadi semakin penting. Tidak hanya konten informasi, namun saluran atau media yang digunakan untuk distribusi informasi juga perlu mendapat pengamanan. Kecanggihan teknologi yang berkembang saat ini memudahkan seseorang atau pihak-pihak tertentu untuk mendapatkan informasi apapun yang diinginkannya. Kemudahan ini sering kali disalahgunakan oleh pihak tidak yang bertanggung jawab dalam melakukan tindakan ilegal seperti peretasan informasi yang bersifat sensitif atau rahasia.

Keamanan sebuah informasi merupakan hal penting yang perlu mendapat perhatian. Salah satu bentuk pengamanan informasi adalah dengan melakukan enkripsi menggunakan metode atau teknik kriptografi. Jenis informasi yang dapat dilindungi dapat berupa teks, gambar ataupun bentuk digital lainnya. Teknik kriptografi digunakan untuk mengamankan atau melindungi suatu pesan/informasi [1], [2]. Informasi dilindungi dari pengguna yang tidak berhak, dalam arti hanya mereka yang memiliki izin/akses saja yang dapat mengetahui isi informasi tersebut [3]. Proses kriptografi dibagi menjadi dua bagian, yaitu proses enkripsi dan proses dekripsi, yang dapat simetris atau asimetris [4].

Salah satu teknik kriptografi untuk pengamanan data adalah Vigenere cipher (VC). Teknik ini digolongkan

[*)] Penulis korespondensi (Rihartanto)
Email: rihartanto@polnes.ac.id

Submitted: 24 August 2019; Revised: 28 April 2020; Accepted: 6 May 2020; Published: 31 July 2020

sebagai kriptografi klasik yang relatif mudah untuk dipecahkan. VC menerapkan subsitusi polialpabetik untuk menyandikan suatu informasi. Versi asli dari VC hanya digunakan untuk menyandikan 26 huruf alphabet. Namun, rentang karakter yang disandikan ini dapat diperluas sesuai dengan tujuan implementasinya. Rentang nilai ini dapat diperluas menjadi 128 karakter untuk mengakomodasi ASCII standar atau menjadi 256 karakter agar dapat mengakomodasi seluruh nilai ASCII standar dan ASCII extended [5], [6]. Rentang nilai hingga 256 ini juga berguna untuk mengakomodasi nilai piksel pada citra [7], [8].

Kelemahan utama dari VC adalah terjadinya repetisi penggunaan kunci dalam proses enkripsi. Sifat penggunaan kunci yang berulang tersebut digunakan dalam uji Kasiski dan uji Friedman untuk memecahkan sandi Vigenere. Uji Kasiski dan uji Friedman digunakan untuk memperkirakan panjang kunci yang digunakan pada VC. Uji Kasiski dilakukan dengan cara memperkirakan kemunculan kata-kata yang berulang, sedangkan uji Friedman menggunakan ketidakrataan frekuensi karakter pada hasil enkripsi. Algoritme lain yang lebih modern yang digunakan untuk memecahkan enkripsi VC adalah algoritme cuckoo search [9] dan algoritme genetika [10].

Beberapa pendekatan sudah dilakukan dalam mengoptimasi VC, di antaranya dengan memperluas rentang karakter, melakukan kombinasi algoritme, implementasi kunci yang berbeda, dan menambah siklus enkripsi. Perluasan rentang karakter dari 26 menjadi 128 dalam [5] dan 256 dalam [6] memberikan hasil berupa susunan acak dari karakter ASCII. Perluasan ini juga memungkinkan VC digunakan untuk melakukan engkipsi pada citra [7], [8]. Namun, implementasi VC pada citra masih memerlukan optimasi lebih lanjut karena VC hanya memberi perubahan pada warna citra. VC pada citra yang dioptimasi dengan *bit circular shift* mampu menghasilkan citra yang secara visual tidak mudah dikenali [7].

Kombinasi VC dengan algoritme lain juga telah diterapkan, di antaranya dengan *double transposition* [11] dan *double transposition cipher with one time pad (OTP) cipher* [6]. Transposisi dilakukan untuk menambahkan karakteristik difusi yang tidak dimiliki oleh VC sehingga hasil enkripsi yang diperoleh dari kombinasi tersebut memenuhi karakteristik konfusi dan difusi dari sebuah enkripsi.

Modifikasi pada kunci yang digunakan pada proses enkripsi dan penambahan siklus enkripsi merupakan upaya lainnya untuk meminimalkan kemunculan deretan karakter yang berulang pada hasil enkripsi dengan VC. Beberapa di antaranya adalah OTP [6], modifikasi kunci secara random [5], kunci berupa citra keabuan [8] dan melakukan enkripsi lebih dari satu kali menggunakan kunci yang berbeda [12]. Pendekatan tersebut terbukti mampu meminimalkan kemunculan karakter berurutan yang merupakan ciri khas dari VC.

Kajian ini menerapkan rotasi matriks bujursangkar untuk mengoptimasi hasil enkripsi VC dalam mengamankan informasi dalam bentuk teks. Metode ini bertujuan untuk melakukan difusi pada hasil konfusi dari VC agar meningkatkan keacakan hasil enkripsi untuk menghilangkan ciri berulang yang merupakan karakteristik dari VC, dan untuk mengetahui ukuran bujursangkar yang mampu menghasilkan entropi optimal.

II. METODE PENELITIAN

Pengamanan teks dalam kajian ini dilakukan dalam beberapa tahapan. Yang pertama adalah operasi enkripsi menggunakan Vigenere cipher 128 (VC-128) yang dilanjutkan dengan rotasi bujursangkar terhadap hasil enkripsi tersebut. VC-128 ini merupakan pengembangan dari algoritme yang sudah ada sebelumnya. Rotasi bujursangkar didasari pada pola perputaran pada salah satu sisi permainan rubik. Implementasi rotasi bujursangkar ini dipengaruhi oleh jumlah karakter yang dirotasi dan ukuran bujursangkar yang digunakan pada setiap rotasi.

A. Vigenere cipher dan Vigenere cipher 128

VC-128 merupakan pengembangan dari algoritme VC standar. Algoritme VC menerapkan operasi modulus untuk mendapatkan sandi dari setiap karakter, atau untuk mendapatkan kembali karakter asli yang sudah disandikan. Persamaan 1 digunakan untuk melakukan enkripsi, sedangkan Persamaan 2 digunakan untuk melakukan dekripsi [13]. Parameter C_i adalah karakter cipher ke-i, P_i adalah karakter pesan ke-i, dan K_i adalah karakter kunci ke-i. Nilai 26 merupakan jumlah maksimal karakter yang dapat ditangani, yaitu jumlah dari huruf A sampai dengan Z yang seluruhnya huruf besar atau seluruhnya huruf kecil.

$$C_i = \left(P_i + K_i\right) mod\, 26 \tag{1}$$

$$B'[n-j-1,i] = B[i,j] \tag{2}$$

Kondisi ini mengakibatkan banyak karakter lain yang seringkali ada dalam sebuah informasi, tidak dapat diakomodasi oleh formula ini. Karakter-karakter tersebut di antaranya adalah angka, tanda baca, kombinasi huruf besar, dan huruf kecil serta karakter simbol yang ada pada keyboard. Karena hal tersebut, maka jangkauan karakter VC perlu diperluas menjadi 128 karakter agar dapat mengakomodasi semua karakter yang terdapat pada keyboard yang merupakan representasi dari 128 karakter ASCII standar. Modifikasi formula VC menjadi VC-128 ditunjukkan pada Persamaan 3 untuk melakukan enkripsi dan Persamaan 4 untuk melakukan dekripsi [5].

$$C_i = \left(P_i + K_i\right) mod\, 128 \tag{3}$$

$$P_i = \left(C_i - K_i\right) mod\, 128 \tag{4}$$

Perbedaan utama VC dengan VC-128 adalah adanya proses filterisasi pada VC yang merupakan proses awal sebelum enkripsi dilakukan. Proses filterisasi bertujuan untuk menghilangkan semua karakter yang tidak dapat

diperoses oleh VC dan mengubah seluruh huruf menjadi huruf besar. Berbeda dengan VC, pada VC-128 seluruh karakter pada teks asli dapat langsung dienkripsi sehingga tidak ada data atau informasi yang hilang. Jika pada VC proses enkripsi dan dekripsi dilakukan berdasarkan posisi huruf dalam urutan abjad, pada VC-128 proses enkripsi dan dekripsi dilakukan berdasarkan nilai ASCII dari karakter yang bersangkutan. Hal ini berakibat pada adanya kemungkinan bahwa hasil enkripsi tidak memiliki simbol yang terlihat secara visual.

B. Rotasi matrik bujursangkar

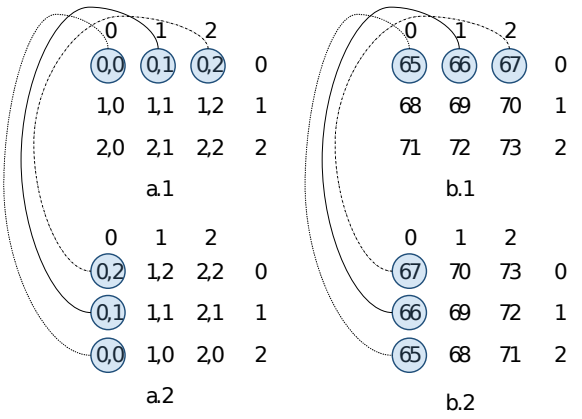
Rotasi matrik bujursangkar digunakan untuk mendapatkan perubahan kedudukan atau posisi elemen dalam matrik bujursangkar dengan cara diputar melalui pusat dan/atau sudut tertentu. Operasi ini diimplementasikan pada larik berbentuk bujursangkar, dimana jumlah baris larik sama dengan jumlah kolomnya. Sebagai pusat rotasi adalah titik tengah bujursangkar. Arah rotasi yang dapat dilakukan adalah searah jarum jam (*clock wise*, CW) atau berlawanan arah jarum jam (*counter clock wise*, CCW). Sedangkan jarak putar dalam satu rotasi adalah perpindahan sejauh 90 derajat.

Ilustrasi rotasi CCW pada larik ditunjukkan pada Gambar 1. Larik dibaca menurut format baris dan kolom. Bagian a.1 dan b.1 menunjukkan koordinat awal dan elemen larik sebelum dilakukan rotasi, sedangkan bagian a.2 dan b.2 menunjukkan hasil perpindahan setelah dilakukan rotasi CCW. Elemen 65 yang semula berada pada posisi [0,0] berpindah ke posisi [2,0]. Elemen 66 yang semula berada pada posisi [0,1] berpindah ke posisi [1,0] dan elemen 67 yang semula berada pada posisi [0,2] berpindah ke posisi [0,0]. Hal yang sama juga dilakukan untuk seluruh elemen lainnya pada larik. Sebagai pusat rotasi adalah bujursangkar, yaitu posisi [1,1].

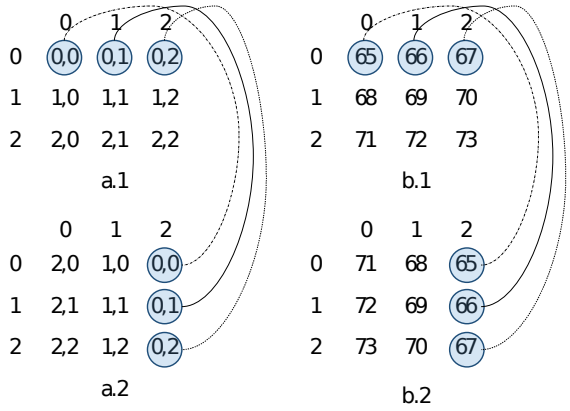
Operasi rotasi CCW pada larik bujursangkar secara matematis dapat dinyatakan pada Persamaan 5. *B* adalah larik sebelum rotasi, sedangkan *B'* adalah larik hasil rotasi, *i* adalah indeks baris dan *j* adalah indeks kolom. Jumlah baris dan jumlah kolom diwakili oleh *n* yang bernilai 3 (Gambar 1). Hasil rotasi CCW untuk B[1,2] adalah B'[0,1] yang diperoleh dari [3-2-1, 1].

$$B'[n-j-1,i]=B[i,j] \quad (5)$$

Gambar 2 memberikan ilustrasi rotasi CW pada larik. Seperti pada rotasi CCW, bagian a.1 dan b.1 menunjukkan koordinat awal dan elemen awal sebelum dilakukan rotasi, sedangkan bagian a.2 dan b.2 menunjukkan hasil perpindahan setelah rotasi dilakukan. Elemen 65 yang semula berada pada posisi [0,0] berpindah ke posisi [0,2]. Elemen 66 yang semula berada pada posisi [0,1] berpindah ke posisi [1,2] dan elemen 67 yang semula berada pada posisi [0,2] berpindah ke posisi [2,2]. Hal yang sama juga dilakukan untuk seluruh elemen lainnya pada larik. Pusat rotasi adalah bujursangkar, yaitu posisi [1,1].



Gambar 1. Rotasi CCW pada larik: (a) koordinat larik, (b) elemen larik



Gambar 2. Rotasi CW pada larik: (a) koordinat larik, (b) elemen larik

Operasi rotasi CW pada larik bujursangkar secara matematis dapat dinyatakan pada Persamaan 6. *B* adalah larik sebelum rotasi, sedangkan *B'* adalah larik hasil rotasi. Jumlah baris dan jumlah kolom pada array diwakili oleh *n* yang bernilai 3. Indeks baris adalah *i* dan indeks kolom adalah *j*. Hasil rotasi CW untuk B[1,2] adalah B'[2,1] yang didapat dari [2, 3-1-1].

$$B'[j,n-i-1]=B[i,j] \quad (6)$$

Penerapan Persamaan 5 dan 6 untuk melakukan rotasi ditunjukkan pada Algoritme 1. Perulangan dilakukan menurut baris dan kolom, dimana perpindahan dilakukan untuk setiap elemen larik sesuai indeksnya masing-masing. Baik rotasi CW maupun CCW, jika dilakukan empat kali berturut-turut, hasilnya adalah sama seperti sebelum dilakukan rotasi. Hasil yang diperolah pada satu kali rotasi CCW sama dengan tiga kali rotasi CW, dan sebaliknya. Hasil dua kali rotasi CCW sama dengan hasil dua kali rotasi CW.

Sebelum operasi rotasi bujursangkar dapat dilakukan, teks yang akan dirotasi terlebih dahulu disusun ke dalam larik berbentuk bujursangkar atau mendekati bujursangkar. Ukuran larik yang dibentuk disesuaikan dengan jumlah karakter yang dimasukkan

ke dalam larik tersebut. Cara menentukan jumlah baris dan kolom larik ditunjukkan pada Algoritme 2.

Jika terdapat tiga buah teks dengan jumlah yang berbeda, masing-masing berjumlah 81, 88 dan 115, maka ukuran jumlah baris dan kolom yang digunakan untuk menampung setiap karakter dari teks tersebut dapat diuraikan sebagai berikut:

- Dengan jumlah karakter 81, maka jumlah baris dan kolom adalah 9 yang merupakan nilai akar dari 81. Larik yang dihasilkan adalah larik bujursangkar;
- Dengan jumlah karakter 88, maka jumlah kolom adalah 9 yang merupakan nilai integer dari akar 88, sedangkan jumlah baris adalah jumlah kolom ditambah satu sehingga menjadi 10 karena kolom lebih kecil dari nilai akar. Larik yang dihasilkan adalah larik yang hampir bujursangkar;
- Dengan jumlah karakter 115, mula-mula jumlah baris dan kolom diberi nilai sama yaitu 10 yang merupakan nilai integer dari akar 115. Jumlah baris kemudian ditambah satu menjadi 11 karena jumlah kolom lebih kecil dari akar 115. Karena hasil perkalian baris dengan kolom (10×11) lebih kecil dari jumlah karakter, maka jumlah kolom ditambah satu sehingga menjadi 11. Larik yang dihasilkan adalah larik bujursangkar 11×11.

Pada kondisi jumlah baris dan jumlah kolom tidak persis sama dengan nilai integer dari akar jumlah karakter, dapat diartikan bahwa jumlah elemen larik adalah lebih besar dari jumlah karakter yang dimasukkan ke dalam larik. Dalam kasus seperti ini, maka pada posisi elemen larik yang kosong tersebut diisi dengan karakter lain yang memiliki nilai ASCII lebih besar dari 127. Hal ini disebabkan karena nilai ASCII 0-127 sudah digunakan pada proses VC-128.

Setelah larik terbentuk, operasi rotasi dapat dilakukan. Ukuran bujursangkar minimal untuk operasi rotasi adalah 2×2 dan ukuran maksimal adalah sesuai jumlah baris atau jumlah kolom yang diambil dari yang paling kecil antara jumlah baris dan jumlah kolom. Gambar 3 menunjukkan urutan rotasi bujursangkar 2×2 pada larik 4×5, dimana (0,0), (0,1) dan seterusnya sampai (3,4) merupakan koordinat elemen pada larik.

Optimasi hasil enkripsi VC-128 menggunakan rotasi bujursangkar ditunjukkan pada Algoritme 3. Pada proses enkripsi VC-128, masukan adalah berupa file teks dan kunci dimasukkan dari keyboard. Teks dapat terdiri dari huruf besar, huruf kecil, angka, tanda baca dan simbol lainnya, termasuk karakter kontrol, seperti enter dan tabulasi. Kunci untuk proses VC-128 dibatasi minimal 6 karakter yang dapat terdiri dari huruf, angka dan simbol. Hal ini bertujuan untuk meningkatkan keacakan hasil enkripsi.

Hasil enkripsi VC-128 menjadi masukan pada proses rotasi bujursangkar. Agar hasil VC-128 dan hasil optimasinya dapat diukur, maka rotasi CCW digunakan pada proses enkripsi dan rotasi CW pada proses dekripsi. Hasil VC-128 yang sudah dioptimasi dengan rotasi bujursangkar merupakan hasil enkripsi akhir yang disimpan kembali sebagai file teks.

Algoritme 1. Rotasi Bujursangkar

Input: array bujursangkar
Output: array bujursangkar

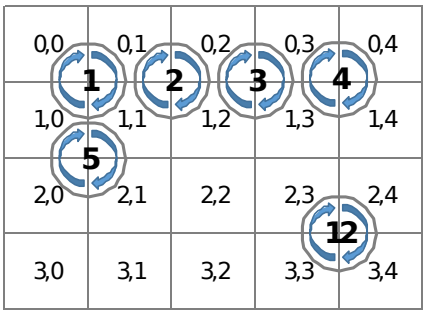
```
Function rotasi_CCW(arr_in)
1:      n ← ambil panjang sisi array
2:          for i ← 0 sampai n-1
3:              for j ← 0 sampai n-1
4:                  arr_out[n-j-1, i] = arr_in[i,j]
                end for
            end for
5:      return arr_out

Function rotasi_CW(arr_in)
6:      n ← ambil panjang sisi array
7:          for i ← 0 sampai n-1
8:              for j ← 0 sampai n-1
9:                  arr_out[j, n-i-1] = arr_in[i,j]
                end for
            end for
10:     return arr_out
```

Algoritme 2. Penentuan ukuran larik untuk menampung hasil enkripsi VC-128

Input: ciphertext
Output: ukuran array

```
  Begin
1:    jml_char ← hitung jumlah karakter ciphertext
2:    akar ← √jml_char
3:    baris, kolom ← int(akar)
4:    if kolom < akar
5:        baris ← kolom + 1
      end if
6:    if baris × kolom < jml_char
7:        kolom ← kolom + 1
      end if
8:    return (baris, kolom)
  End
```



Gambar 3. Urutan rotasi bujursangkar 2×2 pada larik 4×5

Proses dekripsi dilakukan dalam urutan yang berlawanan dengan proses enkripsi. Masukan yang digunakan aadalah file teks hasil enkripsi yang sudah dioptimasi. Dekripsi ini menggunakan rotasi yang berlawanan dengan yang digunakan pada proses

enkripsi. Hasil rotasi menjadi masukan pada proses dekripsi menggunakan kunci yang sesuai. Hasilnya adalah pesan yang sama dengan pesan aslimya.

C. Pengukuran kinerja optimasi VC-128

Pengukuran kinerja hasil optimasi VC-128 menggunakan rotasi bujursangkar dilakukan menggunakan entropi (H) dan efek Avalance (AE). Entropi digunakan untuk mengukur keacakan data [14], sedangkan AE digunakan untuk mengukur seberapa banyak bit data yang berubah. Entropi dihitung menggunakan formula entropi Shannon yang ditunjukkan pada Persamaan 7. Nilai entropi tertinggi yang dapat dicapai pada sebuah citra adalah 8, sementara pada teks yang hanya menggunakan ASCII standar entropi tertinggi yang mungkin diperoleh adalah 7. Semakin tinggi nilai entropi menunjukkan tingkat keacakan yang semakin tinggi.

$$H(x) = -\sum_{i=1}^{n} P(x_i) \log 2\, P(x_i) \qquad (7)$$

Dalam penelitian ini, entropi dihitung berdasarkan besarnya peluang jarak antar karakter dalam teks, bukan dari peluang kemunculan karakter itu sendiri. Hal ini disebabkan karena operasi rotasi bujursangkar berakibat pada perubahan urutan karakter dalam teks, tanpa satu pun mengubah karakter yang ada. Dari hal ini, jika entropi diukur berdasarkan peluang kemunculan karakter, maka diperoleh nilai yang sama, meskipun urutan karakter tersebut mengalami perubahan.

Efek Avalanche digunakan untuk menilai seberapa signifikan perubahan yang terjadi pada cipherteks karena adanya perubahan kecil, baik pada pesan maupun pada kunci. AE dihitung menggunakan Persamaan 8. AE dikatakan baik jika perubahan bit yang terjadi berkisar antara 45 % hingga 60 % [15]. Semakin banyak bit yang berubah mengindikasikan bahwa algoritme enkripsi tersebut semakin sulit untuk dipecahkan.

$$AE = \frac{Jumlah\ bit\ yang\ berubah}{Jumlah\ bit\ cipherteks} \times 100\,\% \qquad (8)$$

III. Hasil dan Pembahasan

Pengujian VC-128 dan optimasinya dengan rotasi bujursangkar menggunakan tiga buah teks uji seperti ditunjukkan pada Tabel 1. Teks pertama adalah teks dalam Bahasa Indonesia, yang kedua teks dalam Bahasa Inggris, dan yang ketiga berisi kata-kata yang diulang-ulang sebanyak jumlah tertentu. Setiap teks memiliki arti secara harfiah, yaitu terdiri dari huruf besar, huruf kecil, angka, tanda baca, dan memiliki jumlah karakter yang relatif sama. Nilai entropi untuk teks asli tersebut berturut-turut adalah 5,6736 untuk teks pertama, 5,7866 untuk teks kedua dan 3,8739 untuk teks ketiga. Hal ini menunjukkan bahwa teks ketiga yang berisi kata-kata berulang memiliki keacakan yang paling rendah.

Algoritme 3. Enkripsi menggunakan VC-128 dan rotasi bujursangkar

Input: plaintext, kunci
Output: ciphertext
Begin
1: teks ← baca file plaintext
2: kunci ← masukkan kunci enkripsi
3: cipher ← enkripsi VC128(teks, kunci)
4: arr ← masukkan setiap karakter cipher ke dalam array berbentuk bujursangkar atau mendekati bujursangkar berukuran (baris, kolom)
5: sisi ← ukuran bujursangkar untuk rotasi

6: **for** i ← 0 sampai baris-sisi+1
7: **for** j ← 0 sampai kolom-sisi+1
8: arr[i:i+sisi, j:j+sisi] = rotasi(arr[i:i+sisi, j:j+sisi])
end for
end for
9: cipher ← susun isi arr ke dalam bentuk teks
10: Simpan cipher ke dalam file ciphertext
End

Kunci untuk melakukan enkripsi dan dekripsi VC-128 adalah `abc123`, yaitu terdiri dari tiga huruf dan tiga angka berurutan yang seluruhnya berjumlah 6 (enam) karakter. Kunci ini digunakan untuk semua teks uji. Hasil enkripsi menggunakan VC-128 ditunjukkan pada Tabel 1 beserta entropi hasil enkripsi. Secara visual, hasil enkripsi terdiri dari karakter dan simbol yang tidak beraturan dan sangat berbeda dari karakter aslinya. Hasil ini relevan dengan kajian dalam [5], [6], [8], [11] dengan teks asli mengalami perubahan signifikan menjadi karakter yang relatif tidak memiliki arti tertentu.

Selain terjadinya perubahan karakter akibat operasi VC-128, keacakan hasilnya pun turut meningkat. Peningkatan entropi untuk setiap teks uji berturut-turut adalah sebesar 0,7113, 0,6317 dan 0,9983. Peningkatan terbesar terjadi pada teks yang berisi kata-kata yang berulang. Sebelum dioptimasi, hasil enkripsi VC-128 terlebih dahulu dimuat ke dalam larik berbentuk bujursangkar atau mendekati bujursangkar dengan mengikuti Algoritme 2. Larik yang digunakan untuk menampung seluruh karakter hasil enkripsi berukuran 18x17 sehingga ukuran bujursangkar yang dapat digunakan untuk melakukan optimasi bervariasi mulai 2×2 hingga 17×17.

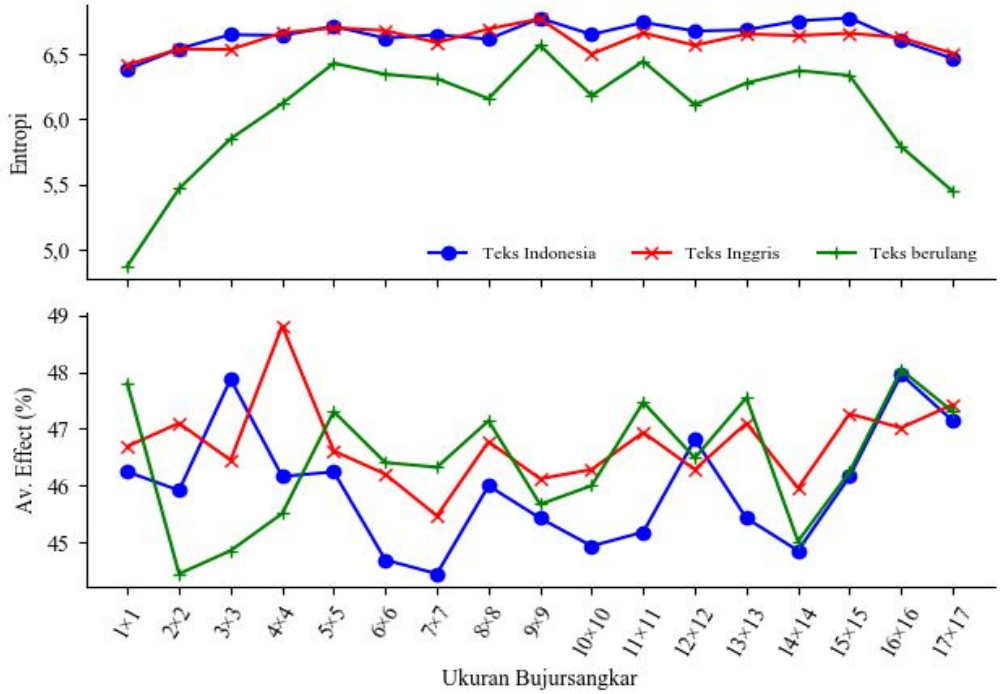
Pengukuran kinerja optimasi menggunakan ukuran bujursangkar yang berbeda ditunjukkan pada Tabel 2. Perbandingan nilai entropi dan AE dari ketiga teks uji tersebut ditunjukkan pada Gambar 4. Hasil tersebut menunjukkan bahwa semakin kecil ukuran bujursangkar, maka semakin banyak jumlah rotasi yang diperlukan. Perpindahan karakter pada tabel tersebut menunjukkan banyaknya perpindahan karakter dari posisi awal menuju posisi tujuan rotasi. Jumlah ini diperoleh dari jumlah karakter sesuai ukuran bujursangkar yang dirotasikan dikalikan dengan jumlah

Tabel 1. Teks uji dan hasil enkripsi dengan VC-128

No	Teks asli	Jumlah karakter	Hasil enkripsi	Entropi enkripsi
1.	Vigenère128 adalah metode untuk menyandikan 128 karakter pada tabel ASCII standar. Tergolong pada subtitusi polialfabetik. Tabel ASCII standar mencakup huruf besar, huruf kecil, angka dan simbol-simbol seperti !@#$%^&*/, dan karakter seperti tab dan enter. Optimasi VC128 dilakukan dgn rotasi bujursangkar.	306	7KJ┬ ←SG¶cjSBFD↔‼←┌OH%!┤F┐X▼&(L┐P┬ ,BPG→↔¶O┐¶cjSLCU↕↔'FT└!‼┤B┐W↕¶↑M┐$┘u│*┐V%‼! ECU_R•FTJ ▲"OI└!‼┤B┐V&¶'JVX $←SQQO→‼▼GCE┬&└L┼└│‼┴FN└r│v*+└$&¶OFD#R FPF↕↔(Q┐K&$(G┐E┬%¶S♬└┼'%VH└└┤┬JN※Q‼! HMDQ┬¶O┐V→▼┴PN┼$← CQOQ%↑QGU%←S┐"−UW◀•↕]R┤BP└└‼ %BMW┬$STGS┬$'J┐W↕¶SECQQ┤! UGU_R┐QVL▲‼&J┐9tce├┐G→▲¶LWN↕ SEIQQ$"UCV→R┴VLX# %¶OIN↕$a	306	6,3849
2.	Information is a valuable asset which needs to be secured. Secure from unauthorized persons or users. According to its form, information can be in the form of text, images, audio, and video. The aim of this research is to secure information stored in the text, using Vigenere cipher and square rotation.	303	*PI $ BVL SJU└↕R)BNX↕¶▼F┐D$%↑U┐Z├←┬I┐Q┬┤┤T┐W R┴F┐V┬┴(SGG_R−FEX#┤SGTR▲R(OCX%→"SK]┬┬SQGU$!! T┐R#R(TGU$`S"EF $┤JPJQ&"┌KW $R├PTP]R└OHR#▼¶UKR▼R┬BP└‼┤SJP└%→↑┌HR#▼SPH└%┤ +U♬└→▼¶HGV]R¶VFL ^SBPGQ(└EGR_R•IG└↕← ┌QIQ&←JU└#┤&FCU¶→SJU└%!STGF&$↑┌KQ┤!%NCW→!!┌UW $↑E┐L▼R'IG└%┤+U♬└&%└OI└•↔→FPH#┤SDKS├┤ %┌CQ┴R&RWD#┤SSQW↕&└PP◀±±±	303	6,4183
3.	Indonesia Raya. Indonesia Raya.	303	*PG ↑TKDQ┘¶ZC◀Q{!EQQ┬%└B┐5↕+¶※┐,▼┬"OGV→‼S3C \↕`S*PG ↑TKDQ┘¶ZC◀Q{!EQQ┬ %└B┐5↕+¶※┐,▼┬"OGV→‼S3C\↕`S*PG ↑TKDQ┘¶ZC◀Q{! EQQ┬%└B┐5↕+¶※┐,▼┬"OGV→‼S3C\↕`S*PG ↑TKDQ┘¶ZC◀Q{!EQQ┬%└B┐5↕+¶※┐,▼┬"OGV→‼S3C\↕`S*PG ↑TKDQ┘¶ZC◀Q{!EQQ┬%└B┐5↕+¶※┐,▼┬"OGV→‼S3C\↕`S*PG ↑TKDQ┘¶ZC◀Q{!EQQ┬%└B┐5↕+¶※┐,▼┬"OGV→‼S3C\↕`S*PG ↑TKDQ┘¶ZC◀±±±	303	4,8722

Tabel 2. Hasil pengukuran satu kali rotasi CCW pada hasil enkripsi VC-128

Bujur sangkar	Jml. Rotasi	Perpindahan huruf	Durasi (detik)			Entropi			Avalance Effect (%)		
			teks1	teks2	teks3	teks1	teks2	teks3	teks1	teks2	teks3
1×1	306	306	0,0021	0,0022	0,0021	6,3849	6,4183	4,8722	46,24	46,69	47,79
2×2	272	1088	0,0024	0,0023	0,0025	6,5424	6,5395	5,4658	45,92	47,10	44,44
3×3	240	2160	0,0028	0,0026	0,0028	6,6502	6,5359	5,8498	47,88	46,45	44,85
4×4	210	3360	0,0032	0,0030	0,0031	6,6442	6,6637	6,1203	46,16	48,82	45,51
5×5	182	4550	0,0033	0,0033	0,0034	6,7139	6,7036	6,4304	46,24	46,61	47,30
6×6	156	5616	0,0037	0,0036	0,0035	6,6229	6,6815	6,3456	44,69	46,20	46,41
7×7	132	6468	0,0038	0,0037	0,0037	6,6493	6,5884	6,3119	44,44	45,47	46,32
8×8	110	7040	0,0039	0,0037	0,0037	6,6161	6,6941	6,1585	46,00	46,77	47,14
9×9	90	7290	0,0038	0,0037	0,0037	6,7768	6,7717	6,5689	45,42	46,12	45,67
10×10	72	7200	0,0037	0,0035	0,0035	6,6510	6,5016	6,1817	44,93	46,28	46,00
11×11	56	6776	0,0033	0,0032	0,0032	6,7448	6,6623	6,4455	45,18	46,94	47,47
12×12	42	6048	0,0029	0,0029	0,0028	6,6781	6,5683	6,1119	46,81	46,28	46,49
13×13	30	5070	0,0024	0,0023	0,0023	6,6878	6,6544	6,2783	45,42	47,10	47,55
14×14	20	3920	0,0018	0,0017	0,0017	6,7558	6,6433	6,3747	44,85	45,96	45,02
15×15	12	2700	0,0013	0,0012	0,0012	6,7790	6,6607	6,3364	46,16	47,26	46,24
16×16	6	1536	0,0007	0,0008	0,0007	6,6028	6,6276	5,7890	47,96	47,02	48,04
17×17	2	578	0,0003	0,0003	0,0002	6,4661	6,5058	5,4458	47,14	47,43	47,30



Gambar 4. Entropi dan AE sesuai ukuran bujursangkar

rotasi, misalnya bujursangkar 4×4 terdiri dari 16 karakter dikalikan dengan jumlah rotasinya (210 rotasi) sehingga jumlah perpindahan posisi sebanyak 3360 kali.

Ukuran bujursangkar sangat berpengaruh pada jumlah rotasi yang dilakukan (Tabel 2). Semakin kecil ukuran bujursangkar, maka semakin banyak jumlah rotasi yang diperlukan. Ukuran bujursangkar dan jumlah rotasi secara langsung berpengaruh pada jumlah perpindahan huruf dan karakter pada larik. Jika ditampilkan dalam bentuk grafik, jumlah perpindahan huruf dan karakter ini membentuk kurva distribusi normal.

Durasi rotasi menunjukkan waktu yang diperlukan setiap ukuran bujursangkar untuk melakukan rotasi mulai dari posisi kiri atas sampai mencapai posisi kanan bawah sesuai urutan yang ditunjukkan pada Gambar 3. Waktu rotasi ini merupakan waktu rata-rata yang diambil dari 60 kali pengujian untuk setiap ukuran bujursangkar. Banyaknya perulangan dalam pengujian ini dilakukan untuk menghindari adanya ganguan karena adanya komputasi lain yang dikerjakan oleh sistem operasi pada waktu yang bersamaan. Namun, pencatatan waktu masih belum mempertimbangkan adanya manajemen memori pada sistem operasi. Jenis komputer yang digunakan untuk pengujian adalah laptop dengan processor Intel Core i7 2,7 GHz, RAM 4 GB, harddisk SATA 5400 rpm, dan menggunakan sistem operasi Windows 7 64 bit. Pemrograman dilakukan menggunakan Python 3.6.

Meski tidak berubah secara linier, peningkatan dan penurunan durasi rotasi dipengaruhi oleh jumlah karakter dalam setiap rotasi. Jumlah rotasi yang tinggi pada bujursangkar berukuran kecil (1×1, 2×2) dan jumlah rotasi yang rendah pada bujursangkar berukuran besar (16×16, 17×17) sama-sama menghasilkan durasi yang singkat jika dibandingkan dengan rotasi pada bujursangkar berukuran 7×7 sampai 10×10. Jumlah rotasi dan jumlah karakter dalam setiap rotasi berpengaruh pada jumlah perpindahan karakter. Secara umum, semakin banyak jumlah perpindahan karakter maka akan semakin lama waktu yang diperlukan untuk menyelesaikan seluruh rotasi.

Secara umum, optimasi menggunakan rotasi bujursangkar berhasil meningkatkan keacakan dari hasil enkripsi seperti ditunjukkan pada Gambar 4. Entropi pada bujursangkar 1x1 adalah entropi hasil enkripsi tanpa rotasi. Hal ini dikarenakan pada bujursangkar 1×1 hanya berisi satu karakter, sehingga operasi rotasi akan menghasilkan karakter itu sendiri. Penggunaan ukuran bujursangkar lainnya meningkatkan keacakan dengan tingkat yang berbeda-beda.

Hasil pengujian menggunakan ketiga data uji tersebut menunjukkan bahwa keacakan hasil optimasi juga dipengaruhi oleh isi dari teks aslinya. Namun, dari ketiga grafik entropi tersebut menunjukkan adanya kemiripan pola, yaitu nilainya cenderung rendah pada bujursangkar berukuran kecil (2×2, 3×3) dan bujursangkar berukuran besar (16×16, 17×17). Rotasi bujursangkar yang menghasilkan nilai entropi yang cenderung tinggi untuk seluruh data terjadi pada bujursangkar berukuran 5×5, 9×9, 11×11, dan 15×15. Entropi tertinggi adalah 6,7790 yaitu pada teks berbahasa Indonesia menggunakan bujursangkar 15×15. Namun, rotasi optimal diperoleh dari penggunaan bujursangkar 9×9 karena seluruh data uji secara bersama-sama menunjukkan nilai entropi yang tinggi.

Peningkatan entropi tertinggi adalah sebesar 34,8 % pada teks berulang dibandingkan dengan teks berbahasa Indonesia dan teks berbahasa Inggris masing-masing 6,1

% dan 5,5 %. Hal ini menunjukkan bahwa rotasi bujursangkar mampu meningkatkan keacakan, terutama untuk data yang awalnya memiliki keacakan yang rendah. Peningkatan keacakan ini menunjukkan kecenderungan yang sama dengan penggunaan bit *circular shift* seperti dalam [7] dan *double transposition* dan OTP dalam [6] dalam meningkatkan hasil enkripsi VC.

Peningkatan keacakan pada teks berbahasa Indonesia dan teks berbahasa Inggris yang sejak awal memang sudah memiliki keacakan yang tinggi, relatif kecil dibandingkan dengan yang terjadi pada teks berulang yang memiliki tingkat keacakan yang jauh lebih rendah. Hasil pengujian ini juga menunjukkan bahwa ukuran bujursangkar optimal adalah bujursangkar yang berukuran setengah dari ukuran bujursangkar maksimal yang dapat digunakan.

Berbeda dengan nilai entropi, hasil pengujian menggunakan AE tidak menunjukkan adanya pola tertentu pada perubahan ukuran bujursangkar yang digunakan untuk optimasi. Secara keseluruhan, nilai AE lebih kecil dari 50 % sehingga kinerja enkripsi dapat dinyatakan baik menurut [15] karena AE berkisar antara 45 % hingga 60 %. AE untuk seluruh data uji dalam kajian ini mendekati 45 % atau lebih besar. Untuk seluruh data uji, penggunaan bujursangkar ukuran 7×7 dan 14×14 memberikan nilai AE yang paling rendah.

Ketepatan hasil dekripsi dibanding dengan teks aslinya tidak ditampilkan dalam penelitian ini. Hal ini didasari konsep bahwa enkripsi dan dekripsi merupakan satu kesatuan proses, dimana hasil enkripsi harus dapat dikembalikan tepat sama sebagaimana aslinya.

IV. KESIMPULAN

Rotasi bujursangkar mampu meningkatkan keacakan hasil enkripsi VC-128. Semakin kecil ukuran bujursangkar, maka semakin banyak jumlah rotasi yang diperlukan. Keacakan optimal untuk semua data uji diperoleh menggunakan rotasi dengan bujursangkar berukuran sedang yaitu bujursangkar 9×9. Peningkatan keacakan tertinggi terjadi pada teks berulang yang pada awalnya memiliki keacakan yang paling rendah. Peningkatan keacakan hasil enkripsi cenderung rendah untuk teks yang awalnya sudah memiliki keacakan yang tinggi. Pengujian menggunakan efek Avalanche menunjukkan bahwa terjadi perubahan berkisar antara 45 % hingga 49 % dari teks aslinya. Besarnya perubahan ini telah cukup memenuhi kriteria sebagai metode enkripsi yang baik.

UCAPAN TERIMA KASIH

Terimakasih disampaikan kepada P3M Politeknik Negeri Samarinda dan Direktorat Jendral Penguatan Riset dan Pengembangan Kementerian Riset, Teknologi dan Pendidikan Tinggi sesuai dengan kontrak nomor 1207/PL7/LK/2019 dan 151/SP2H/LT/DRPM/2019.

DAFTAR PUSTAKA

[1] S. A. Hannan and A. M. A. M. Asif, “Analysis of polyalphabetic transposition cipher techniques used for encryption and decryption,” *International Journal of Computer Science and Software Engineering*, vol. 6, no. 2, pp. 41-46, 2017.

[2] H. Delfs, K. Paterson, and R. Cramer, *Introduction to cryptography: principles and application*, third edition. Berlin: Springer-Verlag GmnH, 2015.

[3] A. E. Handoyo, D. R. I. M. Setiadi, E. H. Rachmawanto, C. A. Sari, and A. Susanto, “Teknik penyembunyian dan enkripsi pesan pada citra digital dengan kombinasi metode LSB dan RSA,” *Jurnal Teknologi dan Sistem Komputer*, vol. 6, no. 1, pp. 37-43, 2018. doi: 10.14710/jtsiskom.6.1.2018.37-43

[4] R. Dixit and K. Ravindranath, “Encryption techniques & access control models for data security : A survey,” *International Journal of Engineering & Technoogy.*, vol. 7, no. 1.5, pp. 107–110, 2018. doi: 10.14419/ijet.v7i1.5.9130

[5] A. Rizal, D. S. B. Utomo, R. Rihartanto, and M. E. Hiswati, “Modified key using multi-cycle key in vigenere cipher,” *International Journal of Recent Technology and Engineering*, vol. 8, no. 2, pp. 2600–2606, 2019. doi: 10.35940/ijrte.B1313.0982S1119

[6] A. Priyam, “Extended vigenère using double transposition cipher with one time pad cipher,” *International Journal of Engineering, Science and Advanced Reaserch*, vol. 1, no. 2, pp. 62–65, 2015.

[7] A. Susanto, T. Khotimah, M. T. Sumadi, J. Warsito, and R. Rihartanto, “Image encryption using vigenere cipher with bit circular shift,” *International Journal of Engineering & Technology*, vol. 7, no. 2.2 (2018), pp. 62–64, 2018. doi: 10.14419/ijet.v7i2.2.12734

[8] I. Saputra, M. Mesran, N. A. Hasibuan, and R. Rahim, “Vigenere cipher algorithm with grayscale image key generator for secure text file,” *International Journal of Engineering Research & Technology*, vol. 6, no. 01, pp. 266–269, 2017.

[9] A. K. Bhateja, A. Bhateja, S. Chaudhury, and P. K. Saxena, “Cryptanalysis of vigenere cipher using cuckoo search,” *Applied Soft Computing*, vol. 26, pp. 315–324, 2015. doi: 10.1016/j.asoc.2014.10.004

[10] A. Bhateja and S. Kumar, “Genetic algorithm with elitism for cryptanalysis of vigenere cipher,” in *International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, Ghaziabad, India, Feb. 2014, pp. 373–377. doi: 10.1109/ICICICT.2014.6781311

[11] N. Sinha and K. Bhamidipati, “Improving security of vigenère cipher by double columnar transposition,” *Int. J. Comput. Appl.*, vol. 100, no. 14, pp. 6–10, 2014, doi: 10.5120/17591-8290

[12] S. K. Mandal and A. R. Deepti, “A cryptosystem based on vigenere cipher by using multilevel encryption scheme,” *International Journal of Computer Science and Information Technologies*, vol. 7, no. 4, pp. 2096–2099, 2016.

[13] S. Rubinstein-Salzedo, “The vigenere cipher,” in *Cryptography*. *Springer Undergraduate Mathematics Series*, Gewerbestrasse: Springer International Publishing AG, 2018. doi: 10.1007/978-3-319-94818-8

[14] T. O. Kvålseth, “On the measurement of randomness (uncertainty): A more informative entropy,” *Entropy*, vol. 18, no. 5, pp. 1–15, 2016. doi: 10.3390/e18050159. doi: 10.3390/e18050159

[15] Sugiyanto and R. K. Hapsari, “Pengembangan algoritma advanced encryption standard pada sistem keamanan SMS berbasis Android menggunakan algoritma vigenere,” *Ultimatics: Jurnal Teknik Informatika*, vol. 8, no. 2, pp. 131–138, 2016. doi: 10.31937/ti.v8i2.528