ENKRIPSI TEKS DENGAN PENDEKATAN BITSTREAM

Ansar Rizal[1], Andi Azhari[2], Arief Susanto[3], Sugiyamto[4], Eko Nur Wahyudi[5]
[1,2]Teknik Komputer, Jurusan Teknologi Informasi, Politeknik Negeri Samarinda
[3]Teknik Informatika, Fakultas Teknik, Universitas Muria Kudus
[4]Sistem Informasi, Fakultas Teknologi Informasi dan Industri, Universitas Stikubank Semarang
[5]Manajemen Informatika, Fakultas Vokasi, Universitas Stikubank Semarang
e-mail: [1]ansardeuy@gmail.com, [2]andiazhari0318@gmail.com, [3]arief.susanto@umk.ac.id,
[4]sugiyamtogik@edu.unisbank.ac.id, [5]eko@edu.unisbank.ac.id

ABSTRAK

Informasi merupakan aset penting baik itu bagi pemerintah, organisasi swasta, universitas, LSM, atau perorangan. Perkembangan teknologi yang pesat menjadikan informasi semakin penting lagi. Karenanya informasi perlu mendapat pengamanan. Bukan hanya isinya, tetapi saluran atau media yang digunakan untuk penyebaran informasi. Pada penelitian ini pendekaran bitstream digunakan pengaman data teks menggunakan kunci simetris sederhana. Penggunaan operasi XOR dalam penggabungan datastream dan keystream menghasilkan konfusi yang tinggi pada ciphertext. Hasil penelitian menunjukkan terjadinya perubahan yang signifikan pada hasil enkripsi yang ditunjukkan oleh nilai Avalanche effect yang tinggi dan nilai koefesien korelasi yang rendah. Nilai Avalanche effect tertinggi adalah 53.91% dan koefesien korelasi terbaik adalah -0.0094. Hasil ini membuktikan bahwa enkripsi dengan pendekaran bitstream dapat menghasilkan ciphertext yang baik, dimana hasil enkripsi tidak dipengaruhi oleh teks aslinya.

Kata Kunci: *pendekatan bitstream, xor, konfusi, avalanche effect*.

1. PENDAHULUAN

Informasi merupakan aset penting baik itu bagi pemerintah, organisasi swasta, universitas, LSM, atau perorangan. Perkembangan teknologi yang pesat menjadikan informasi semakin penting lagi. Karenanya informasi perlu mendapat pengamanan. Bukan hanya isinya, tetapi saluran atau media yang digunakan untuk penyebaran informasi. Penggunaan internet yang luas semakin memudahkan seseorang atau pihak tertentu untuk mendapatkan informasi apapun yang diinginkannya. Kemudahan akses ini membuka peluang terjadinya penyalahgunaan oleh pihak yang tidak bertanggung jawab untuk melakukan tindakan ilegal seperti peretasan data sensitif atau rahasia.

Keamanan informasi merupakan aspek penting yang membutuhkan perhatian serius. Upaya pengamanan informasi diantaranya dengan penyandian menggunakan metode atau teknik tertentu. Sedangkan jenis informasi yang dapat diamankan tidak hanya berupa teks tetapi juga gambar atau bentuk digital lainnya.

Kriptografi merupakan seni atau ilmu yang digunakan untuk mengamankan atau melindungi data dan informasi [1][2]. Tujuan pengamanan informasi adalah untuk mengamankannya dari pengguna yang tidak berhak, dalam konteks bahwa hanya mereka yang memiliki izin yang sesuai yang dapat mengakses konten informasi[3]. Proses kriptografi dibagi menjadi dua bagian, yaitu proses enkripsi dan dekripsi. Kedua proses tersebut biasanya membutuhkan kata kunci, dimana kata kunci tersebut bisa simetris atau asimetris[4] tergantung dari teknik kriptografi yang digunakan.

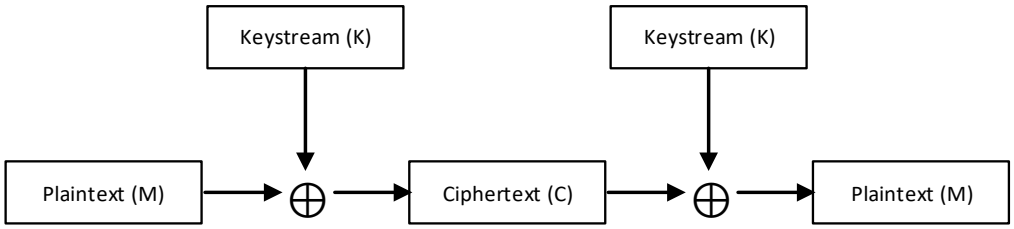
Dilihat dari sisi jumlah data yang diproses, kriptografi dapat dikelompokkan menjadi block cipher dan stream cipher. Stream cipher adalah algoritma yang mengenkripsi 1 bit atau byte data dalam satu waktu. Algoritma ini menggunakan aliran bit pseudorandom tak terbatas sebagai kuncinya. Agar implementasi stream cipher tetap aman, generator pseudorandom harus tidak dapat diprediksi dan kuncinya tidak boleh digunakan kembali. Stream cipher dirancang untuk mendekati cipher yang ideal, yang dikenal sebagai One-Time Pad[5].

Dalam penelitian ini, pengaman data teks dilakukan dengan pendekaan bitstream menggunakan kunci tetap. Sesuai dengan karakteristik dari stream cipher, enkripsi dengan pendekatan bitstream bertujuan untuk menhasilkan data terenkripsi yang memiliki konfusi relatif tinggi.

2. METODE PENELITIAN

2.1 Stream Cipher

Stream cipher termasuk dalam kelompok algoritma enkripsi kunci simetris di mana setiap bit pada plainteks (data stream) digabungkan dengan bit kunci (keystream). Keystream biasanya merupakan bit-bit yang hasilkan dari pseudorandom. Penggabungan ini dilakukan satu per satu, satu bit pada setiap waktu. Operasi yang umum digunakan untuk penggabungan dalam stream cipher adalah operasi XOR. Penggabungan tersebut akan menghasilkan bit ciphertext. Representasi diagram untuk struktur operasi pada stream cipher ditunjukkan pada Gambar 1.



Gambar 1. Struktur Stream Cipher

Operasi XOR digunakan untuk menggabungkan setiap bit atau byte dari pesan dengan bit atau byte pada keystream pada urutan yang bersesuaian untuk menghasilkan ciphertext. Demikian juga untuk melakukan dekripsi, operasi XOR dilakukan pada ciphertext menggunakan keystream yang sama. Pada struktur ini pseudorandom menghasilkan 8 bit keystream[6] untuk digabungkan dengan 1 byte dari plaintext. Misalkan keystream yang dihasilkan adalah 01101100 dan plaintext selanjutnya adalah 11001100 maka proses untuk menghasilkan ciphertext adalah

```
   11001100   plaintext
⊕  01101100   keystream
   10100000   ciphertext
```

Demikian pula, proses untuk mendapatkan kembali pesan yang dienkripsi adalah

```
   10100000   ciphertext
⊕  01101100   keystream
   11001100   plaintext
```

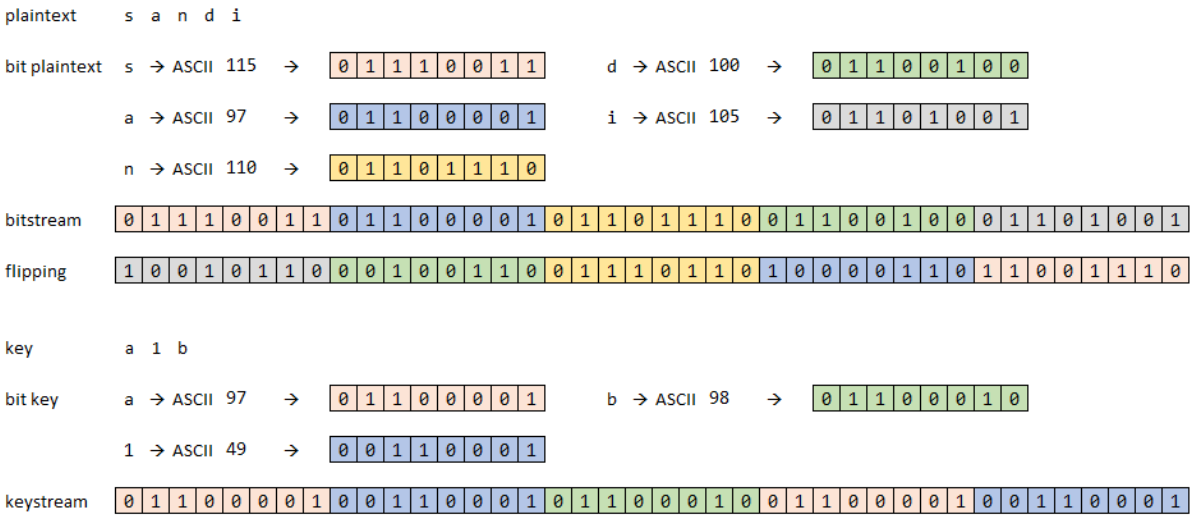
Notasi matematis untuk proses enkripsi dan dekripsi ditunjukkan oleh Persamaan (1) dan persamaan (2).

$$C = P \oplus \mathrm{K} \quad (1)$$

$$P = C \oplus \mathrm{K} \quad (2)$$

2.2 Pendekatan bitstream

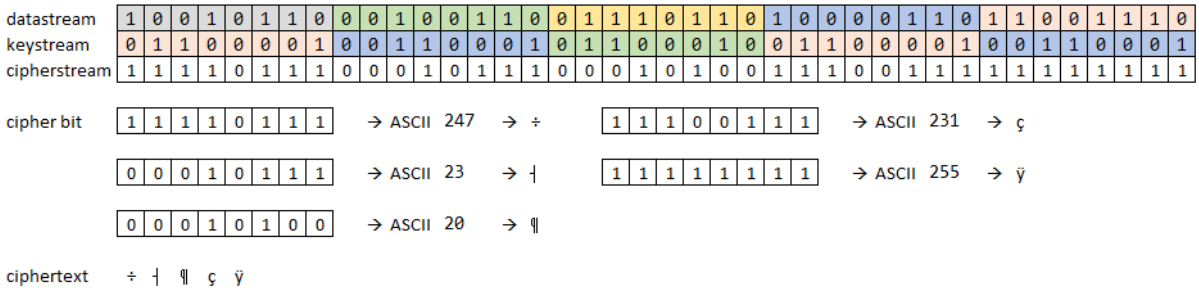
Pendekatan bitstream dalam penelitian ini meniru cara kerja pada stream cipher dimana operasi penggabungan menggunakan operasi XOR. Perbedaannya adalah pada pembentukan datastream keystream. Datastream adalah untaian bit dari setiap karakter pada plainteks yang susun secara berurutan. Selanjutnya dilakukan pencerminan horizontal pada untaian bit tersebut untuk mendapakan urutan bit yang berkebalikan dari untaian bit aslinya. Pembentukan bitstream dari teks yang akan dienkripsi dan pementukan bitstream kunci yang digunakan pada proses enkripsi ditunjukkan pada Gambar 2.



Gambar 2. Ilustrasi pembentukan datastream dan keystream

Misalkan teks yang akan dienkripsi adalah “sandi” menggunakan kunci “a1b”. Setiap karakter pada teks terlebih dahulu dikembalikan ke dalam nilai ASCII untuk selanjutnya direpresentasikan dalam 8 bit biner. Bit biner dari setiap karakter disatukan menjadi bitstream yang merupakan untaian dari keseluruhan bit dari teks tadi. Selanjutnya dilakukan operasi pencerminan atau flipping sehingga bit terakhir menjadi bit pertama sementara bit pertama menjadi bit terakhir, proses ini dilakukan pada seluruh untaian bit. Hasil flipping merupakan datastream yang akan dienkripsi.

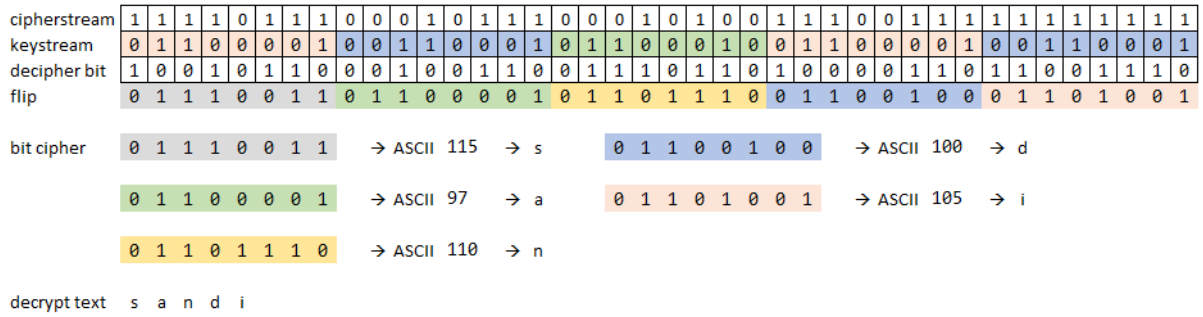
Mirip dengan karakter teks, karakter kunci juga diubah menjadi untaian bit panjang berdasarkan nilai ASCII dari setiap karakternya. Panjang bitstream kunci ini disesuaikan dengan panjang data stream dengan cara mengulangi penggunaan karakter pada kunci seperti yang dilakukan pada Vigenere cipher. Dalam hal ini bitstream untuk karakter keempat pada teks akan dipasangkan dengan karakter pertama kunci, karakter kelima pada teks dipasangkan dengan karakter kedua pada kunci dan seterusnya jika masih ada karakter berikutnya. Setelah diperoleh panjang keystream yang sama dengan panjang datastream selanjutnya dilakukan proses enkripsi menggunakan operator XOR yang ditunjukkan pada Gambar 3.



Gambar 3. Operasi XOR untuk enkripsi

Datastream dan keystream yang diperoleh selanjutnya digabungkan menggunakan operasi XOR untuk mendapatkan cipherstream. Operasi XOR hanya akan bernilai 1 jika bit pada datastream berbeda dengan bit pada keystream. Selanjutnya setiap 8 bit cipherstream diambil untuk dikembalikan ke nilai ASCII untuk mendapatkan karakter hasil enkripsi.

Pada proses dekripsi seperti ditunjukkan pada Gambar 4, operasi XOR dilakukan terhadap cipherstream menggunakan keystream yang sama dengan yang digunakan pada proses enkripsi. deciperbit merupakan bitstream yang dihasilkan dari penggabungan cipherstream dengan keystream. Selanjutnya dilakukan flipping, setelah itu bitstream hasil flipping diambil per 8bit untuk diambil nilai ASCII-nya guna mendapatkan kembali karakter hasil dekripsi.



Gambar 4. Operasi XOR untuk dekripsi

Dari kedua ilustrasi pada Gambar 3 dan Gambar 4 terlihat bahwa proses enkripsi dan dekripsi menggunakan pendekatan bitstream ini sesuai dengan Persamaan (1) dan Pesamaan (2) pada stream cipher. Algoritma untuk kedua proses ini ditunjukkan pada Algortima 1 untuk proses enkripsi dan Algoritma 2 untuk proses dekripsi.Terlihat pada kedua algoritma tersebut bahwa operasi bit yang dilakukan adalah sama, hanya saja dalam urutan yang berkebalikan.

Algoritma 1: Proses enkripsi
Input: plaintext, key
Output: ciphertext

```
1   Function bitEncrypt(plaintext, key)
2     dstream ← ubah plaintext menjadi untain bit
3     dstream ← flip(dstream)
4     kstream ← ubah key menjadi untaian bit
5
6     for i ← 0 to len(dstream)
7         cstream[i] = XOR(dstream[i], kstream[i])
8     end for
9
10    ciphertext ← ubah cstream menjadi karakter
11  return ciphertext
```

Algoritma 2: Proses dekripsi
Input: ciphertext, key
Output: deciphertext

```
1.  Function bitDecrypt(ciphertext, key)
2.    cstream ← ubah ciphertext menjadi untain bit
3.    kstream ← ubah key menjadi untaian bit
4.
5.    for i ← 0 to len(cstream)
6.        dstream[i] = XOR(cstream[i], kstream[i])
7.    end for
8.
9.    dstream = flip( dstream )
10.   deciphertext ← ubah dstream menjadi karakter
11. return deciphertext
```

3. HASIL DAN PEMBAHASAN

Pengukuran kinerja hasil enkripsi dengan pendekatan bitstream dilakukan dengan menggunakan Avalanche Effect (AE) dan koefisien korelasi. AE digunakan untuk menilai seberapa signifikan perubahan yang terjadi pada ciphertext karena perubahan kecil pada pesan dan kuncinya. AE dihitung menggunakan Persamaan (3). AE dikatakan baik jika perubahan bit yang terjadi lebih besar dari 45% [7] dan sangat baik jika lebih besar dari 50% [8], [9]. Semakin banyak bit yang berubah, menindikasikan bahwa algoritma enkripsi semakin sulit untuk dipecahkan.

$$AE = \frac{jumlah\ bit\ yang\ berubah\ pada\ ciphertext}{jumlah\ bit\ dalam\ ciphertext} \times 100\% \quad (3)$$

Koefisien korelasi menilai keacakan hasil enkripsi dalam hal ini dengan menilai hubungan antara plaintext dan ciphertext. Koefisien korelasi yang mendekati nol atau kurang dari 0,2 menunjukkan hubungan yang sangat lemah antara plaintext dan ciphertext. Sebaliknya jika nilainya mendekati 1 atau -1 artinya hasil enkripsi sangat dipengaruhi oleh plaintext yang diberikan.

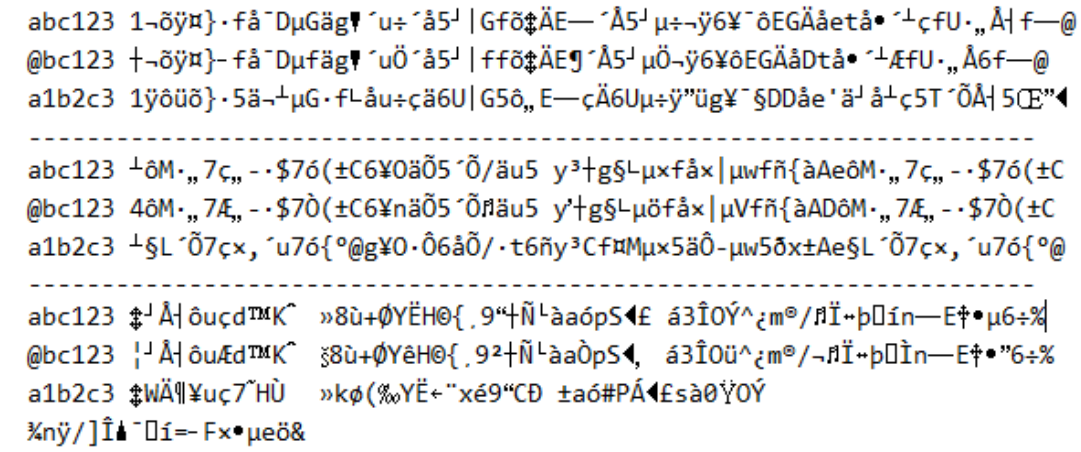
Tiga teks berbeda yang digunakan untuk menguji enkripsi menggunakan pendekatan bitstream ditunjukkan pada Tabel 1. Setiap teks berisi 64 karakter namun memiliki karakteristik yang berbeda. Teks pertama merupakan kalimat biasa, teks kedua terdiri dari frase berulang sementara teks ketiga terdiri dari karakter berurutan pada tabel ASCII. Teks kedua dan ketiga digunakan untuk tujuan pengujian saja, karena dalam kejadian sehari-hari jenis teks seperti ini akan sangat jarang ditemui atau digunakan.

Setiap teks dienkripsi menggunakan tiga buah kunci yang berbeda untuk menilai pengaruh kunci pada setiap karakteristik teks. Kunci yang digunakan adalah abc123, @bc123 dan a1b2c3. Ketiganya merupakan kunci sederhana yang relatif mudah ditebak sehingga digolongkan sebagai kunci yang lemah.Tujuannya tentu saja untuk melihat apakah dengan menggunakan kunci yang lemah dapat memberikan hasil enkripsi yang baik.

Tabel 1. Teks Uji

Fileteks	Plainteks
text1.txt	New nomal adalah kondisi sosial ekonomi dll akibat adanya krisis
text2.txt	NKRI harga mati NKRI harga mati NKRI harga mati NKRI harga mati.
text3.txt	()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg

Setiap teks dienkripsi menggunakan ketiga kunci yang berbeda tersebut. Teks hasil enkripsi ditunjukkan pada Gambar 5 dan hasil pengujiannya ditunjukkan pada Tabel 2. Tiga baris pertama pada Gambar 5 adalah hasil enkripsi kalimat normal, tiga baris berikutnya adalah hasil enkripsi frasa berulang dan tiga baris terakhir adalah hasil enkripsi untuk karakter berurutan. Seluruh hasil enkripsi menunjukkan perubahan yang sangat signifikan dari teks aslinya. Bahkan untuk teks2 yang berisi frasa berulang, hasil enkripsinya tidak menunjukkan adanya perulangan.



Gambar 5. Teks hasil enkripsi menurut kunci yang digunakan

Penggunaan kunci abc123 dan @bc123 yang hanya memiliki perbedaan satu karakter awalnya saja, meskipun hasilnya sangat berbeda dari teks aslinya namun memiliki kemripan yang sangat tinggi. Hal ini terutama disebabkan oleh pola berulang penggunaan karakter kunci agar memiliki panjang keystream yang sama dengan datastreamnya. Sedangkan penggunaan kunci a1b2c3 yang secara keseluruhan memiliki karakter yang sama dengan kunci abc123, menghasilkan ciphertext yang sama sekali berbeda. Hal ini menunjukkan bahwa perbedaan urutan karakter pada kunci memberikan perubahan hasil enkripsi yang lebih besar dibandingkan dengan penggantian beberapa karakter pada kunci.

Tabel 2. Hasil Pengujian

Fileteks	Kunci	Avalanche. Effect	Koef. Korelasi
text1.txt	abc123	0.4962	-0.1616
text1.txt	@bc123	0.5000	-0.1910
text1.txt	a1b2c3	0.5115	-0.1178
text2.txt	abc123	0.5215	0.1061
text2.txt	@bc123	0.5371	0.0853
text2.txt	a1b2c3	0.5391	0.0190
text3.txt	abc123	0.5059	-0.0094
text3.txt	@bc123	0.4863	-0.0217
text3.txt	a1b2c3	0.5039	0.0146

Hasil pengujian menunjukkan bahwa enkripsi teks dengan pendekatan bitstream memberikan hasil enkripsi yang sangat baik, dibuktikan dengan nilai AE yang lebih besar dari 50% untuk sebagian besar hasil enkripsi [8], [9] dan tidak ada nilai AE yang lebih kecil dari 45% [7]. Nilai AE ini jauh lebih tinggi dibanding [10] yang menggunakan kombinasi algoritma vigenere cipher dan rotasi bujursangkar yang termasuk dalam kelompok block cipher. Demikian pula nilai koefesien korelasinya yang berada pada rentang lebih besar dari -0.2 hingga lebih kecil dari 0.2 yang menunjukkan hasil enkripsi memiliki keacakan yang tinggi. Teks hasil engkripsi tidak memiliki korelasi atau memiliki korelasi yang sangat lemah dengan teks aslinya.

Nilai AE yang tinggi serta nilai koefesien korelasi yang rendah mengindikasikan bahwa penggunaan pendekatan bitstream untuk enkripsi teks mampu memberikan hasil enkripsi yang kuat sehingga akan sulit untuk dipecahkan. Keuntungan lainnya adalah algoritma ini cukup sederhana sehingga mudah untuk dimplementasikan. Demikian pula, pendekaan bitsream ini dapat juga digunakan untuk melakukan penyandian pada data digital lainnya selain teks.

Kelemahan dari algoritma yang digunakan pada penlitian ini adalah pada penggunaan memori dalam proses enkripsi dan dekripsi. Dimana panjang pesan yang dienkripsi atau didekripsi akan secara linier meningkatkan penggunaan jumlah memori ketika proses enkripsi atau dekripsi dilakukan.

4. KESIMPULAN

Penelitian ini menunjukkan bahwa pendekatan bitstream untuk enkripsi teks mampu memberikan hasil enkripsi yang baik. Pengujian menggunakan tiga buah teks yang memiliki karakteristik berbeda seluruhnya memberikan nilai AE yang tinggi dan nilai koefesien korelasi yang rendah. Kedua nilai ini mengindikasikan bahwa teks hasil enkripsi akan sulit untuk dipecahkan. Penelitian selanjutnya akan diarahkan pada efesiensi penggunaan sumberdaya komputasi pada implementasi pendekatan bitstream untuk ukuran datas yang besar.

DAFTAR PUSTAKA

[1] S. A. Hannan and A. M. A. M. Asif, “Analysis of Polyalphabetic Transposition Cipher Techniques used for Encryption and Decryption,” *Int. J. Comput. Sci. Softw. Eng.*, vol. 6, no. 2, pp. 41–46, 2017.

[2] H. Delfs and H. Knenbl, *Introduction to Cryptography: Principles and Application*, Third Edit. Berlin: Springer-Verlag GmbH, 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," *J. Teknol. dan Sist. Komput.*, vol. 6, no. 1, p. 37, 2018.

[4] R. Dixit and K. Ravindranath, "Encryption techniques & access control models for data security : A survey," *Int. J. Eng. Technol.*, vol. 7, no. 1.5, pp. 107–110, 2018.

[5] B. Schneier, *Applied Crypthography*, 20th Anniv. Indianapolis: John Wiley & Sons, Inc, 2015.

[6] W. Stallings, *Cryptography and Network Security: Principles and Practice*, Seventh Ed. Harlow: Pearson Education Limited, 2017.

[7] H. Noura, L. Sleem, M. Noura, M. M. Mansour, A. Chehab, and R. Couturier, "A new efficient lightweight and secure image cipher scheme," *Multimed. Tools Appl.*, vol. 77, no. 12, pp. 15457–15484, 2018.

[8] S. D. Mohammed and T. M. Hasan, "Cryptosystems using an improving hiding technique based on latin square and magic square," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 20, no. 1, pp. 510–520, 2020.

[9] J. N. B. Salameh, "A new symmetric-key block ciphering algorithm," *Middle East J. Sci. Res.*, vol. 12, no. 5, pp. 662–673, 2012.

[10] R. Rihartanto, R. K. Ningsih, A. F. O. Gaffar, and D. S. B. Utomo, "Implementation of vigenere cipher 128 and square rotation in securing text messages," *J. Teknol. dan Sist. Komput.*, vol. 8, no. 3, pp. 201–209, 2020.