DESIGNING A SQUARE TRANSPOSITION ALGORITHM WITH A SPIRAL SCHEMATIC

PERANCANGAN ALGORITMA SQUARE TRANSPOSISI DENGAN SKEMA SPIRAL

Hendrik Patiung[1], Alz Danny Wowor[2]
Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana, Salatiga, Jawah Tengah, Indonesia
Email: 672018308@student.uksw.edu1[1] ,*alzdanny.wowor@uksw.edu* [2]

Abstract - This study aims to improve the security of the square transposition algorithm by implementing a spiral scheme as a more complex character position randomization method. Conventional square transposition algorithms have limitations in the variety of encryption patterns, which can reduce their effectiveness in protecting data from pattern-based cryptanalysis attacks. The spiral scheme is implemented by determining the order of placement and retrieval of characters in the transposition box in a circular manner, both from the outside to the inside and from the center point out, following a non-linear pattern. This method is designed to produce a low correlation between plaintext and ciphertext, thereby increasing the level of randomness in the encryption process. In this study, the effectiveness of the spiral scheme is evaluated through statistical analysis and cryptanalysis tests to measure the level of randomness and security of the encryption results compared to the linear transposition pattern. The test results show that the spiral scheme successfully increases the complexity of the ciphertext structure, making it more resistant to pattern analysis-based attacks. The implementation of this spiral scheme is expected to make a significant contribution to the development of more secure encryption techniques, especially for applications that require high protection against digital security threats. Thus, this method offers a stronger alternative in improving data security without requiring high computational complexity.

Keywords - Cryptography, Square Transposition, Spiral.

Abstrak - Penelitian ini bertujuan untuk meningkatkan keamanan algoritma transposisi persegi melalui penerapan skema spiral sebagai metode pengacakan posisi karakter yang lebih kompleks. Algoritma transposisi persegi konvensional memiliki keterbatasan dalam variasi pola enkripsi, yang dapat mengurangi efektivitasnya dalam melindungi data dari serangan kriptanalisis berbasis pola. Skema spiral diterapkan dengan menentukan urutan penempatan dan pengambilan karakter dalam kotak transposisi secara melingkar, baik dari luar ke dalam maupun dari titik pusat keluar, mengikuti pola yang tidak linier. Metode ini dirancang untuk menghasilkan korelasi rendah antara plainteks dan cipherteks, sehingga meningkatkan tingkat keacakan dalam proses enkripsi. Dalam penelitian ini, efektivitas skema spiral dievaluasi melalui analisis statistik dan uji kriptanalisis untuk mengukur tingkat keacakan dan keamanan hasil enkripsi dibandingkan dengan pola transposisi linier. Hasil pengujian menunjukkan bahwa skema spiral berhasil menambah kerumitan struktur cipherteks, sehingga lebih tahan terhadap serangan berbasis analisis pola. Implementasi skema spiral ini diharapkan dapat memberikan kontribusi signifikan dalam pengembangan teknik enkripsi yang lebih aman, terutama bagi aplikasi yang membutuhkan proteksi tinggi terhadap ancaman keamanan digital. Dengan demikian, metode ini menawarkan alternatif yang lebih kuat dalam meningkatkan keamanan data tanpa memerlukan kompleksitas komputasi yang tinggi.

Kata Kunci - Kriptografi, Square transposisi, Spiral.

I. PENDAHULUAN

Di era komputer dan internet saat ini, keamanan data menjadi aspek penting yang perlu diperhatikan, terutama dalam pertukaran data melalui jaringan yang rentan terhadap serangan siber. Kriptografi merupakan ilmu dan seni yang menggunakan enkripsi dan dekripsi untuk memastikan bahwa hanya pihak yang berwenang dapat mengakses data yang terlindungi. Salah satu cara untuk menjaga kerahasiaan informasi adalah melalui penggunaan algoritma kriptografi. Salah satu algoritma yang dikembangkan untuk tujuan ini adalah algoritma transposisi, yaitu teknik enkripsi yang mengubah urutan karakter dalam pesan asli (plaintext) untuk menghasilkan pesan terenkripsi (ciphertext). Algoritma ini tidak mengubah karakter itu sendiri, melainkan hanya posisi karakter, sehingga kunci yang benar dibutuhkan untuk mengembalikan pesan ke bentuk aslinya.

Square Transposisi dengan skema spiral adalah metode yang menggunakan pola spiral untuk menentukan posisi karakter dalam pesan terenkripsi. Spiral terdiri dari kotak-kotak yang diatur dalam baris dan kolom, di mana pola ini digunakan untuk memandu penempatan karakter selama proses enkripsi dan dekripsi. Dengan skema ini, kompleksitas pola penempatan dapat ditingkatkan, sehingga memperkuat tingkat keamanan algoritma transposisi tersebut. Sebagai bagian dari pengujian algoritma yang dirancang, akan diuji seberapa baik skema ini dalam menghasilkan bilangan acak melalui pengujian keacakan (mono bit, block bit, dan run test). Pengujian lainnya adalah uji visualisasi, yang berguna untuk mengetahui seberapa baik luaran bit terdistribusi secara merata pada koordinat Cartesius. Pengujian terakhir adalah uji enkripsi, yang bertujuan melihat seberapa baik kunci yang diperoleh dapat memutus hubungan statistik antara plainteks dan cipherteks [1], [2].

Penelitian ini fokus pada pengembangan dan pengujian algoritma transposisi berbasis skema spiral dengan tujuan meningkatkan keamanan data melalui keacakan yang lebih baik. Ruang lingkup penerapan algoritma ini lebih diutamakan pada data berukuran kecil hingga menengah, seperti kata sandi dan pesan teks. Algoritma ini diuji untuk mendukung aplikasi yang memerlukan keacakan tinggi tanpa menambah kompleksitas teknis pada proses dekripsi. Penelitian ini memiliki batasan dalam hal skala data besar, di mana peningkatan ukuran data dapat menyebabkan keterbatasan performa. Hal ini menambah signifikansi studi karena memberikan solusi yang relevan dan aplikatif di tengah meningkatnya kebutuhan keamanan digital dalam kehidupan sehari-hari, [3], [4], [5] .

Keunikan algoritma ini terletak pada kombinasi transposisi dalam bentuk spiral dan pengujian ketat terhadap keacakan data, yang mencakup uji mono bit, block bit, dan run test. Hasil pengujian menunjukkan bahwa algoritma ini mencapai tingkat keacakan yang tinggi dan mempertahankan korelasi rendah antara plainteks dan cipherteks, memberikan keamanan tambahan terhadap serangan yang berbasis analisis statistik atau kriptanalisis. Algoritma ini dirancang untuk menjaga efisiensi tanpa menambah beban komputasi yang signifikan, memungkinkan implementasi yang cocok untuk aplikasi keamanan ringan, seperti perlindungan data pribadi pada aplikasi mobile atau komunikasi singkat.

Tujuan dari penelitian ini adalah untuk mengembangkan dan menguji efektivitas metode enkripsi berbasis algoritma transposisi dengan menggunakan skema spiral yang dapat meningkatkan keamanan algoritma transposisi dan menilai bagaimana metode skema spiral

dapat berkontribusi pada pengamanan data. Diharapkan penelitian ini akan membantu dalam pengembangan teknik kriptografi yang lebih canggih dan efektif untuk melindungi informasi pribadi di dunia digital yang terus berkembang[6].

II. SIGNIFIKANSI STUDI

A. *Square Transposisi*

Square Tranposition terdiri dari dua proses yaitu memasukan bit ke dalam *square* dan proses pengambilan bit dengan ukuran yang telah di tentukan sebelumnya. Dimisalkan T = input teks, t_i = karakter teks ke-i, dan a_i = karakter biner ke-i, maka:

$$T = \{t_1, t_2, \cdots, t_n\}; \quad n|8, n \in Z^+ \qquad (1)$$

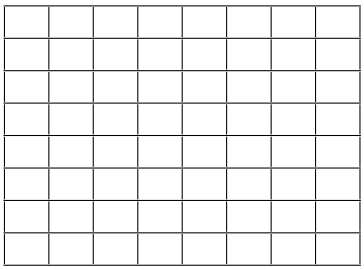
dimana $t_1 = \{a_{01}, a_{02}, a_{03}, \cdots, a_{08}\}$, $t2 = \{a09, a10, a11, \cdots, a16\}$, $t3 = \{a17, a18, a19, \cdots, a24\}$, $\cdots$, $tn = \{a_8n–7, a_8n–6, a_8n–5, \cdots, a_8n\}$.

Jika $n \nmid 8$, maka padding dilakukan sebanyak k, sehingga akan membentuk Persamaan 2.

$$T = \{t_1, t_2, \cdots, t_n, t_n+1, t_n+2, \cdots, t_n+k\} \qquad (2)$$

untuk $(n + k)|8; \quad k = 1, 2, \cdots, 7.$

Square yang digunakan sebagai media transposisi dapat disesuaikan dengan ukuran bit pada input teks. Penelitian ini memilih input teks berukuran 64-bit, sehingga akan square berukuran 8×8, yang ditunjukkan pada Gambar 1.



Gambar 1. *Square* Tranposition 8 × 8.

Skema pemasukkan adalah cara menempatkan setiap bit a_i; $i \in Z_{64}^+$ dalam entri pada *square* dengan aturan tertentu. Misalkan setiap bit setelah dimasukan ke dalam square adalah urutan bit yang diberikan pada Persamaan 3.

$$T_{sq} = \{a_1^*, a_2^*, a_3^*, \cdots, a_{64}^*\} \qquad (3)$$

Skema pengambilan merupakan cara untuk mengambil setiap bit a_i^*, $i \in Z_{64}^+$ dari square dengan aturan tertentu. Notasi setiap bit yang diambil dari *square* $(a_{i\,(j)}^*)$; $\exists\, i,j \in Z_{64}^+$ dimana i adalah indeks pemasukan dan j indeks pengambilan. Persamaaan 4 adalah dataset skema pengambilan $L = \{l_1, l_2, l_3, \cdots, l_8\}$

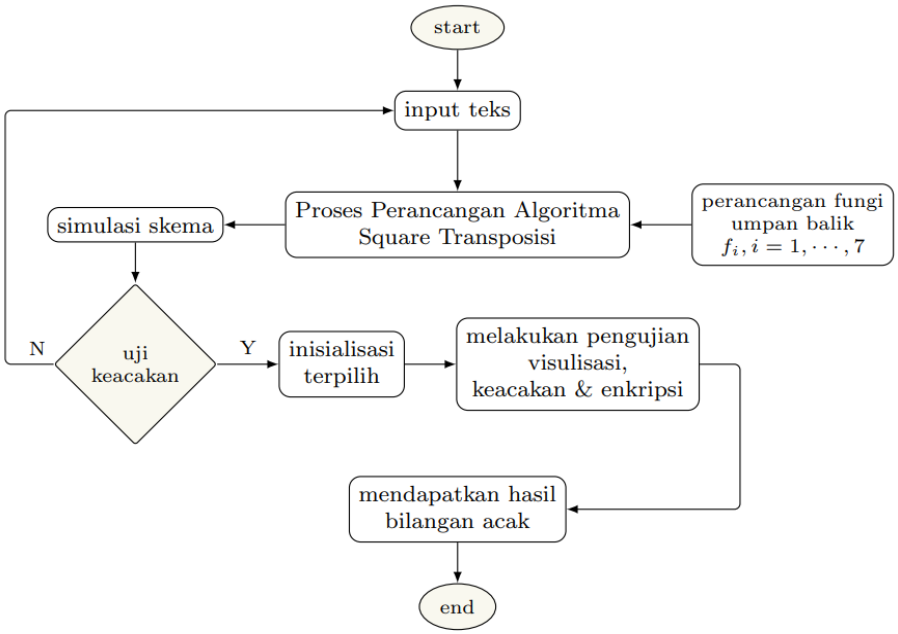
$$\begin{aligned} l_1 &= \{a_{x\,(01)}^*, a_{x\,(02)}^*, a_{x\,(03)}^*, \cdots, a_{x\,(08)}^*\}, \\ l_2 &= \{a_{x\,(09)}^*, a_{x\,(10)}^*, a_{x\,(11)}^*, \cdots, a_{x\,(16)}^*\}, \ldots \quad \ldots \\ l_8 &= \{a_{x\,(57)}^*, a_{x\,(58)}^*, a_{x\,(59)}^*, \cdots, a_{x\,(64)}^*\}. \end{aligned} \qquad (4)$$

dimana$\exists\ x \in Z_{64}^{+}$.

B. Rancangan Penelitian

Alur kerja penelitian secara umum dijelaskan pada Gambar 2 menunjukkan langkah-langkah dari proses perancangan algoritma *square* transposisi hingga menghasilkan bilangan acak. Tahap pertama dimulai dengan input teks, yang kemudian dilanjutkan ke proses perancangan algoritma *square* transposisi. Pada tahap ini, fungsi umpan balik f_i, $i = 1, \cdots, 7$ dirancang untuk digunakan dalam proses algoritma tersebut. Setelah perancangan algoritma selesai, dilakukan simulasi skema untuk menguji efektivitas desain yang dihasilkan.

Setelah simulasi, dilakukan uji keacakan untuk menentukan apakah hasil dari algoritma tersebut memenuhi kriteria keacakan yang diharapkan. Jika hasilnya tidak memenuhi (N), maka proses kembali ke simulasi skema untuk melakukan perbaikan. Namun, jika uji keacakan berhasil (Y), proses berlanjut ke inisialisasi yang dipilih. Setelah itu, dilakukan pengujian visualisasi, keacakan, dan enkripsi untuk memastikan bahwa algoritma bekerja dengan baik. Akhirnya, setelah semua pengujian berhasil, diperoleh hasil berupa bilangan acak, yang menandai akhir dari alur kerja penelitian ini.



Gambar 2. Proses Penelitian

III. HASIL DAN PEMBAHASAN

Berdasarkan Persamaan 1, digunakan 64-bit sebagai input dan square berukuran 8 × 8. Dipilih dua skema pemasukan dengan nilai indeks yang dipilih secara acak, kedua skema secara spiral diberikan pada Gambar 3 dan Gambar 4.

*a*57	*a*18	*a*11	*a*27	*a*12	*a*40	*a*34	*a*51
*a*21	*a*54	*a*20	*a*61	*a*24	*a*62	*a*64	*a*59
*a*38	*a*13	*a*42	*a*05	*a*48	*a*37	*a*06	*a*55
*a*05	*a*45	*a*33	*a*43	*a*15	*a*56	*a*53	*a*31
*a*14	*a*23	*a*07	*a*35	*a*50	*a*39	*a*32	*a*63
*a*10	*a*36	*a*49	*a*58	*a*26	*a*02	*a*22	*a*41
*a*19	*a*30	*a*28	*a*60	*a*17	*a*47	*a*01	*a*46
*a*08	*a*25	*a*52	*a*16	*a*03	*a*29	*a*09	*a*44

Gambar 3. Skema 1 Lotre

*a*63	*a*57	*a*47	*a*33	*a*32	*a*46	*a*56	*a*62
*a*58	*a*48	*a*34	*a*19	*a*18	*a*31	*a*45	*a*55
*a*49	*a*35	*a*20	*a*09	*a*08	*a*17	*a*30	*a*44
*a*36	*a*21	*a*10	*a*03	*a*02	*a*07	*a*16	*a*29
*a*37	*a*22	*a*11	*a*04	*a*01	*a*06	*a*15	*a*28
*a*50	*a*38	*a*23	*a*12	*a*05	*a*14	*a*27	*a*43
*a*59	*a*51	*a*39	*a*24	*a*13	*a*26	*a*42	*a*54
*a*64	*a*60	*a*52	*a*40	*a*25	*a*41	*a*53	*a*61

Gambar 4. Skema 2 Spiral

A. Rancangan Skema Pengambilan

Skema pengambilan adalah sebuah aturan mengambil setiap bit dari dalam square, yang sebelumnya sudah terdapat bit karena proses pemasukan bit. Berikut beberapa skema pengambilan yang digunkan sebagai pasangan dari skema pemasukan. Setting 5 mm untuk bagian kiri menjorok kedalam

1. Skema Pengambilan Horizontal Kiri-Kanan

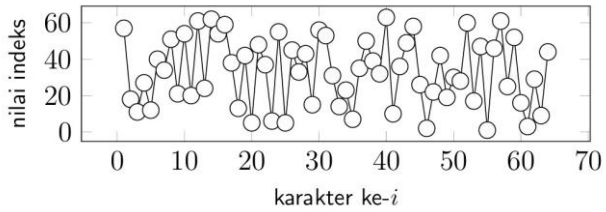
Rancangan ini menggunakan Skema Pemasukan-1 untuk memasukan bit ke dalam square, seperti yang diberikan pada Gambar 5. Proses pengambilan secara horizontal dilakukan dari pojok kiri atas *square* ke kanan. Urutan setiap bit a_{8i+1} untuk $i = 0, 1, \cdots, 7$ selalu berada sebelah kiri dari entri pertama setiap baris ke-$(i + 1)$ *square*.

*a*57 (01)	*a*18 (02)	*a*11 (03)	*a*27 (04)	*a*12 (05)	*a*40 (06)	*a*34 (07)	*a*51 (08)
*a*21 (09)	*a*54 (10)	*a*20 (11)	*a*61 (12)	*a*24 (13)	*a*62 (14)	*a*64 (15)	*a*59 (16)
*a*38 (17)	*a*13 (18)	*a*42 (19)	*a*05 (20)	*a*48 (21)	*a*37 (22)	*a*06 (23)	*a*55 (24)
*a*04 (25)	*a*45 (26)	*a*33 (27)	*a*43 (28)	*a*15 (29)	*a*56 (30)	*a*53 (31)	*a*31(32)
*a*14 (33)	*a*23 (34)	*a*07 (35)	*a*35 (36)	*a*50 (37)	*a*39 (38)	*a*32 (39)	*a*63 (40)
*a*10 (41)	*a*36 (42)	*a*49 (43)	*a*58 (44)	*a*26 (45)	*a*02 (46)	*a*22 (47)	*a*41 (48)
*a*19 (49)	*a*30 (50)	*a*28 (51)	*a*60 (52)	*a*17 (53)	*a*47 (54)	*a*01 (55)	*a*46 (56)
*a*08 (57)	*a*25 (58)	*a*52 (59)	*a*16 (60)	*a*03 (61)	*a*29 (62)	*a*09 (63)	*a*44 (64)

Gambar 5. Skema Pengambilan Horizontal Kiri-Kanan

Skema pengambilan horizontal kiri-kanan dimulai dari a_{57} berdasarkan indeks $j = 1$ sampai $j = 64$ untuk a_{44}. Sehingga diperoleh luaran *Square* Transposition berdasarkan *byte*, seperti

yang diberikan sebelumnya pada Persamaan 6, $L = \{l_1, l_2, l_3, \cdots, l_8\}$ dimana $l_1 = \{a_{57}, a_{18}, \cdots, a_{51}\}$, $l_2 = \{a_{21}, a_{54}, a_{20}, \cdots, a_{59}\}$, $\cdots$, $l8 = \{a08, a25, a52, \cdots, a44\}$.
Hasil transposisi dari Skema Pengambilan-1 dan Skema Pemasukan Horizontal kiri-kanan dapat divisualisasikan dalam koordinat Cartesius, dimana setiap indeks pengambilan (i) sebagai absis dan indeks pemasukan (j) sebagai ordinat. Hasil pengambilan bit secara lengkap diberikan pada Gambar 6.



Gambar 6. Grafik Skema Pengambilan Horizontal Kiri-Kanan

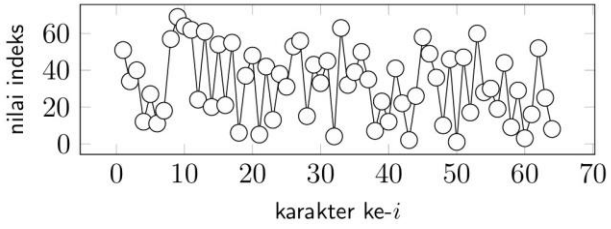
2. Skema Pengambilan Horizontal Kanan-kiri
Rancangan ini menggunakan Skema Pemasukan-1 untuk memasukan bit ke dalam *square*, seperti yang diberikan pada Gambar 7. Proses pengambilan secara horizontal dilakukan dari pojok kanan atas *square* ke kiri. Urutan setiap bit a_{8i+1} untuk $i = 0, 1, \cdots, 7$ selalu berada sebelah kanan dari entri pertama setiap baris ke-$(i + 1)$ *square*.

a57 (08)	a18 (07)	a11 (06)	a27 (05)	a12 (04)	a40 (03)	a34 (02)	a51 (01)
a21 (16)	a54 (15)	a20 (14)	a61 (13)	a24 (12)	a62 (11)	a64 (10)	a59 (09)
a38 (24)	a13 (23)	a42 (22)	a05 (21)	a48 (20)	a37 (19)	a06 (18)	a55 (17)
a04 (32)	a45 (31)	a33 (30)	a43 (29)	a15 (28)	a56 (27)	a53 (26)	a31(25)
a14 (40)	a23 (39)	a07 (38)	a35 (37)	a50 (36)	a39 (35)	a32 (34)	a63 (33)
a10 (48)	a36 (47)	a49 (46)	a58 (45)	a26 (44)	a02 (43)	a22 (42)	a41 (41)
a19 (56)	a30 (55)	a28 (54)	a60 (53)	a17 (52)	a47 (51)	a01 (50)	a46 (49)
a08 (64)	a25 (63)	a52 (62)	a16 (61)	a03 (60)	a29 (59)	a09 (58)	a44 (57)

Gambar 7. Skema Pengambilan Horizontal Kanan-Kiri

Skema pengambilan horizontal kanan-kiri dimulai dari a_{51} berdasarkan indeks $j = 1$ sampai $j = 64$ untuk a_{08}. Sehingga diperoleh luaran *Square* Transposition berdasarkan *byte*, seperti yang diberikan sebelumnya pada Persamaan 4, $L = \{l_1, l_2, l_3, \cdots, l_8\}$ dimana $l_1 = \{a_{51}, a_{34}, \cdots, a_{57}\}$, $l_2 = \{a_{59}, a_{64}, a_{62}, \cdots, a_{21}\}$, $\cdots$, $l8 = \{a44, a09, a29, \cdots, a08\}$.

Hasil transposisi dari Skema Pemasukan-1 dan Skema Pengambilan Horizontal kanan-kiri dapat divisualisasikan dalam koordinat Cartesius, dimana setiap indeks pengambilan (i) sebagai absis dan indeks pemasukan (j) sebagai ordinat. Hasil pengambilan bit secara lengkap diberikan pada Gambar 8.



Gambar 8. Grafik Skema Pengambilan Horizontal Kanan-Kiri

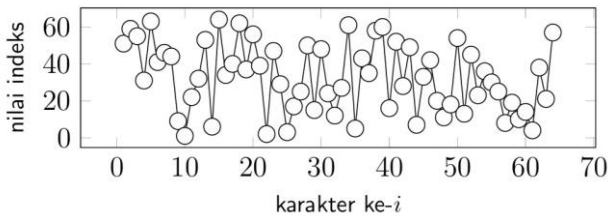
3. Skema Pengambilan Vertikal Atas-Bawah

Square Transposition ini juga menggunakan skema Pemasukan-1 untuk mengisi setiap entri dari square. Pengambilan dilakukan secara vertikal dari arah atas ke bawah, dimulai pada entri pojok kanan atas sampai bagian kanan bawah dari square. Secara umum, setiap bit $a_{i\ (j)}$ dan indeks pengambilan $j = (8z + 1)$; $z \in \{0, 1, \cdots, 7\}$. Jika z genap maka pengambilan dilakukan vertikal atas ke bawah, dan z ganjil maka pengambilan akan dilakukan dari bawah ke atas.

a57 (64)	a18 (49)	a11 (48)	a27 (33)	a12 (32)	a40 (17)	a34 (16)	a51 (01)
a21 (63)	a54 (50)	a20 (47)	a61 (34)	a24 (31)	a62 (18)	a64 (15)	a59 (02)
a38 (62)	a13 (51)	a42 (46)	a05 (35)	a48 (30)	a37 (19)	a06 (14)	a55 (03)
a04 (61)	a45 (52)	a33 (45)	a43 (36)	a15 (29)	a56 (20)	a53 (13)	a31(04)
a14 (60)	a23 (53)	a07 (44)	a35 (37)	a50 (28)	a39 (21)	a32 (12)	a63 (05)
a10 (59)	a36 (54)	a49 (43)	a58 (38)	a26 (27)	a02 (22)	a22 (11)	a41 (06)
a19 (58)	a30 (55)	a28 (42)	a60 (39)	a17 (26)	a47 (23)	a01 (10)	a46 (07)
a08 (57)	a25 (56)	a52 (41)	a16 (40)	a03 (25)	a29 (24)	a09 (09)	a44 (08)

Gambar 9. Skema Pengambilan Vertikal Atas-Bawah

Hasil skema pengambilan vertikal atas-bawah dimulai dari a_{51} berdasarkan indeks $j = 1$ sampai $j = 64$ untuk bit a_{57}. Sehingga luaran *Square* Transposition berdasarkan *byte*, dapat dilihat Gambar 9. $L = \{l_1, l_2, l_3, \cdots, l_8\}$, diamana $l_1 = \{a_{51}, a_{59}, a_{31}, \cdots, a_{44}\}$, $l_2 = \{a_{09}, a_{01}, a_{22}, \cdots, a_{34}\}$, $\cdots$, $l_8 = \{a_{08}, a_{19}, a_{10}, \cdots, a_{57}\}$. Hasil pengambilan berdasarkan bit yang diberikan pada Gambar10.



Gambar 10. Grafik Skema Pengambilan Vertikal Atas-Bawah

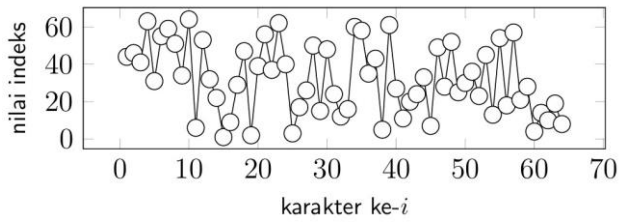
4. Skema Pengambilan Vertikal Bawah-Atas

Square Transposition ini juga menggunakan skema Pemasukan-1 untuk mengisi setiap entri dari square. Pengambilan dilakukan secara vertikal dari arah bawah ke atas, dimulai pada entri pojok kanan bawah sampai bagian kanan atas dari square. Secara umum, setiap bit $a_{i\ (j)}$ dan indeks pengambilan $j = (8z + 1)$; $z \in \{0, 1, \cdots, 7\}$. Jika z genap maka pengambilan dilakukan vertikal atas ke bawah, dan z ganjil maka pengambilan akan dilakukan dari bawah ke atas.

a57 (57)	a18 (56)	a11 (41)	a27 (40)	a12 (25)	a40 (24)	a34 (09)	a51 (08)
a21 (58)	a54 (55)	a20 (42)	a61 (39)	a24 (26)	a62 (23)	a64 (10)	a59 (07)
a38 (59)	a13 (54)	a42 (43)	a05 (38)	a48 (27)	a37 (22)	a06 (11)	a55 (06)
a04 (60)	a45 (53)	a33 (44)	a43 (37)	a15 (28)	a56 (21)	a53 (12)	a31(05)
a14 (61)	a23 (52)	a07 (45)	a35 (36)	a50 (29)	a39 (20)	a32 (13)	a63 (04)
a10 (62)	a36 (51)	a49 (46)	a58 (35)	a26 (30)	a02 (19)	a22 (14)	a41 (03)
a19 (63)	a30 (50)	a28 (47)	a60 (34)	a17 (31)	a47 (18)	a01 (15)	a46 (02)
a08 (64)	a25 (49)	a52 (48)	a16 (33)	a03 (32)	a29 (17)	a09 (16)	a44 (01)

Gambar 11. Skema Pengambilan Vertikal Bawah-Atas

Hasil pengambilan berdasarkan bit yang diberikan pada Gambar 12. Hasil skema pengambilan vertikal bawah-atas dimulai dari a_{44} berdasarkan indeks j = 1 sampai j = 08 untuk bit a_{08}. Sehingga luaran Square Transposition berdasarkan byte L = {l_1, l_2, l_3, ⋯ , l_8}, diamana l_1 = {a_{44}, a_{46}, a_{41}, ⋯ , a_{51}}, l_2 = {a_{34}, a_{64}, a_{06}, ⋯ , a_{09}}, ⋯ , l_8 = {a_{57}, a_{15}, a_{38}, ⋯ , a_{08}}. Visualisasi indeks transposisi dari Skema Pemasukan-1 dan Skema Pengambilan Vertikal bawah-atas diberikan pada Gambar 11.



Gambar 12. Grafik Skema Pengambilan Vertikal Atas-Bawah

B. Pengujian Keacakan pada Nilai Indeks

Metode yang digunakan dalam pengujian keacakan adalah Uji Frekuensi Mono Bit, Blok Bit, dan Run Test, dengan α = 0, 01. Setiap nilai indeks metode transposisi dinyatakan acak apabila dua atau tiga hasil pengujian mempunyai p-value > α. Hasil pengujian secara lengkap diberikan pada Tabel 1. Kombinasi setiap skema pemasukan dan pengambilan dilakukan untuk melihat seberapa baik pasangan skema dirancang atau dipilih, sehingga Square Transposition dapat menghasilkan nilai indeks yang acak.

TABEL I. HASIL PENGUJIAN KEACAKAN SETIAP SKEMA

No	Skema Pemasukan	*p-value*			Hasil
		Mono Bit	Block Bit	Run-Test	
1	Pemasukan-1 & Skema Horizontal Kiri-Kanan	1	0,99996	5,65874E-48	Acak
2	Pemasukan-1 & Skema Horizontal Kanan-Kiri	1	0,99988	5,65874E-48	Acak
3	Pemasukan-1 & Skema Vertikal Atas bawah	1	1	5,65874E-48	Acak
4	Pemasukan-1 & Skema Vertikal Bawah Atas	0,689156517	0,19915	4,11842E-48	Acak
5	Pemasukan-1 & Skema Zig-Zag	1	1	5,65874E-48	
6	Pemasukan-1 & Skema Bajak Sawah Kiri-Kanan	0,689156517	1	3,55781E-22	acak
7	Pemasukan-1 & Skema Bajak Sawah Kanan-Kiri	0,689156517	1	1,18351E-24	acak
8	Pemasukan-1 & Alur tanam Padi		1	1,03E-22	acak
9	Pemasukan-2 & Skema Horizontal Kiri-Kanan	0,689156517	1	4,90289E-47	acak
10	Pemasukan-2 & Skema Horizontal Kanan-Kiri	0,548506236	1	4,74445E-47	acak

11	Pemasukan-2 & Skema Vertikal Atas bawah	0,841480581	1	5,0594E-47	acak
12	Pemasukan-2 & Skema Vertikal Bawah Atas	0,841480581	1	5,0594E-47	tidak acak
13	Pemasukan-2 & Skema Zig-Zag	0,689156517	1	4,90289E-47	acak
14	Pemasukan-2 & Skema Bajak Sawah Kiri-Kanan	0,841480581	1	4,85388E-48	tidak acak
15	Pemasukan-2 & Skema Bajak Sawah Kanan-Kiri	0,689156517	1	4,11842E-48	acak
16	Pemasukan-2 & Alur tanam Padi	1	0,19915	5,65874E-48	acak

Skema Pemasukan-1 & Horizontal Kiri-Kanan, Pemasukan-1 & Horizontal Kanan-Kiri, Pemasukan-1 & Vertikal Atas-Bawah, dan Pemasukan-1 & Skema Vertikal Bawah-Atas memiliki nilai *p-value* rata-rata paling baik karena semua p-value pada uji Mono Bit dan Blok Bit adalah 1, yang merupakan nilai maksimal dalam pengujian keacakan. Ini menunjukkan bahwa skema-skema ini memberikan hasil yang paling acak dan karenanya dianggap paling baik di antara semua skema yang diuji.

Skema Pemasukan-1 & Vertikal Bawah-Atas memiliki nilai *p-value* rata-rata yang cukup rendah, terutama pada uji Blok Bit dengan nilai 0.1991. Pada Skema Pemasukan-2 & Alur Tanam Padi juga memiliki nilai p-value yang rendah pada uji Blok Bit, yaitu 0.1991, meskipun uji Mono Bit menunjukkan keacakan yang baik dengan *p-value* sebesar 1. Ini menunjukkan bahwa meskipun hasilnya acak pada tingkat bit individual, pada level blok, keacakan ini tidak sepenuhnya konsisten. Hal ini menunjukkan bahwa skema ini menghasilkan hasil yang kurang baik dalam hal keacakan, terutama ketika keacakan diukur pada blok-blok data.

C. Pengujian Korelasi

Berisi hasil pembahasan dan bisa perbandingan dari hasil penelitian sebelumnya. Nilai korelasi (r) dapat digunakan untuk melihat seberapa besar hubungan antara input (x) dan output (y) dari algoritme yang berelasi secara statistik. Interval korelasi $-1 \leqslant r \leqslant 1$, dan apabila r mendekati 0 , maka algoritme mampu membuat input-output tidak berhubungan secara statistik. dalam kondisi ini, jika r < 0 maka nilai mutlak |r| dapat digunakan untuk mengetahui jarak r 0. Hasil pengambilan berdasarkan bit yang diberikan pada Tabel 2.

TABLE II. HASIL PENGUJIAN KORELASI INPUT-OUTPUT

No	Metode Transposisi	Nilai korelasi |*r*|			Mean
		Magelang	**xxxxxxxy**	$Em4r@n9	
1	Pemasukan-1 & Skema Horizontal Kiri-Kanan	0,151	0,076	0,268	0,165
2	Pemasukan-1 & Skema Horizontal Kanan-Kiri	-0,481	-0,814	-0,38	-0,558
3	Pemasukan-1 & Skema Vertikal Atas bawah	0,827	-0,292	-0,09	0,148
4	Pemasukan-1 & Skema Vertikal Bawah Atas	-0,503	-0,008	0,185	-0,109
5	Pemasukan-1 & Skema Zig-Zag	0,247	-0,131	-0,464	-0,116

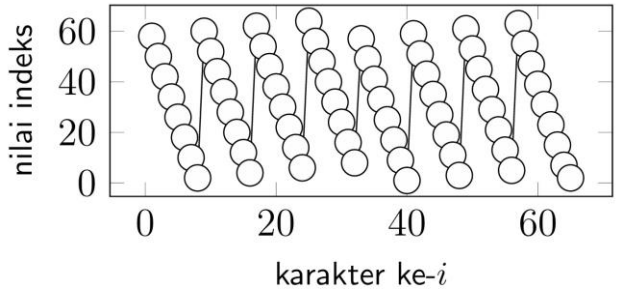
6	Pemasukan-1 & Skema Bajak Sawah Kiri-Kanan	-0,252	-0,118	0,205	-0,055
7	Pemasukan-1 & Skema Bajak Sawah Kanan-Kiri	0,081	0,171	-0,316	-0,021
8	Pemasukan-1 & Alur tanam Padi	-0,131	0,118	-0,473	-0,162
9	Pemasukan-2 & Skema Horizontal Kiri-Kanan	-0,082	0,757	0,058	0,244
10	Pemasukan-2 & Skema Horizontal Kanan-Kiri	-0,745	0,129	-0,227	-0,281
11	Pemasukan-2 & Skema Vertikal Atas bawah	-0,024	0,168	-0,259	-0,038
12	Pemasukan-2 & Skema Vertikal Bawah Atas	-0,178	-0,549	0,177	-0,183
13	Pemasukan-2 & Skema Zig-Zag	0,522	0,136	-0,386	0,091
14	Pemasukan-2 & Skema Bajak Sawah Kiri-Kanan	-0,56	-0,265	-0,061	-0,295
15	Pemasukan-2 & Skema Bajak Sawah Kanan-Kiri	0,154	0,232	-0,653	-0,089
16	Pemasukan-2 & Alur tanam Padi	0,569	-0,311	-0,769	-0,170
17	DES	0,342	0,126	0,374	0,342
18	AES	0,376	0,429	0,277	0,376

Pengujian korelasi digunakan tiga input plainteks, diharapkan dapat mewakilkan variasi teks yang mungkin digunakan oleh user. Input “magelang”, untuk mewakilkan input teks biasa, karena biasanya user menggunakannya. Pengujian kedua yang lebih ekstrim adalah input yang sama, digunakan “xxxxxxxy” (tidak digunakan “yyyyyyyyy”, karena rumus korelasi menjadi tidak terdefenisi). Pengujian ketiga, adalah “$Em4r@n9” juga mewakilkan variasi simbol, angka, dan huruf yang digunakan sebagai input.

Hasil pengujian pada Tabel 2, menunjukkan transposisi DES dan AES mempunyai nilai rata-rata korelasi lebih besar dibandingkan dengan Square Transposition. Bila di lihat jarak dari 0, sebagai acuan sebuah algoritme dapat menyamarkan input, maka setiap skema dengan Square Transposisition lebih baik dalam membuat input dan output tidak berhubungan secara statistik.

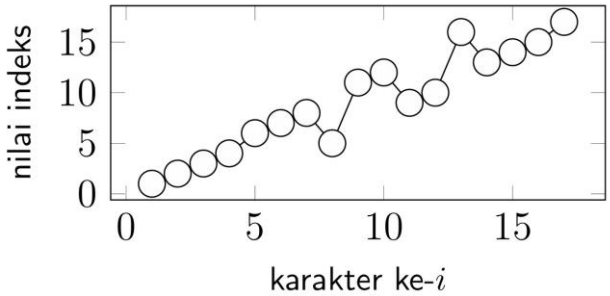
D. Visuallisasi Nilai Indeks

Nilai indeks transposisi dari DES pada Gambar 13 memperlihatkan pola yang konsisten. Luaran 64- bit dari DES selalu terdiri dari 8-bit. Setiap kelompok 8-bit ini mengikuti pola yang berulang, setiap nilai indeks menghasilkan delapan pola, di mana nilai indeks dalam kelompok pertama menurun secara bertahap dari nilai tertinggi hingga nilai terendah. Misalkan a_i dimana (i = 1, 2, 3, ⋯ , 8) sebagai nilai indeks pada kelompok pertama, maka nilai pada posisi yang sama di kelompok berikutnya selalu menjadi $a_i + 1$ dalam (mod 64).



Gambar 13. Grafik Nilai Indeks Transposisi DES

Transposisi pada AES juga menunjukkan pola pada nilai indeksnya, seperti yang terlihat pada Gambar 14. Nilai indeks AES terbagi menjadi kelompok-kelompok yang masing-masing terdiri dari empat karakter. Kelompok pertama (4 – 0),terdapat empat histogram yang meningkat tanpa ada histogram yang berbeda. Kelompok kedua memiliki pola (3 – 1), di mana tiga histogram meningkat dan satu histogram memiliki nilai yang berbeda. (2 – 2) dan (1 – 3) Pola ini berlanjut untuk kelompok ketiga dan kelompok keempat.



Gambar 14. Grafik Nilai Indeks Transposisi AES

Masalah yang terjadi pada DES dan AES, ketika urutan atau pola pada data {1, 2, 3, ⋯ , n} diketahui, maka peluang yang besar untuk menemukan data {n + 1, n + 2, ⋯ }. Hal ini akan membuat algoritme semakin lemah. Penebakan atau pencarian data nilai indeks Square Tranposition sulit dilakukan, dan kriptanalis memerlukan waktu yang lebih lama untuk dapat mencari relasi input dan output.

IV. KESIMPULAN

Penelitian ini berhasil mengembangkan algoritma transposisi berbasis skema spiral yang efektif dalam meningkatkan keamanan data. Berdasarkan pengujian keacakan menggunakan mono bit, block bit, dan run test, algoritma ini mencapai nilai p-value sebesar 0,999 pada beberapa skema, menunjukkan tingkat keacakan yang sangat tinggi dan kesulitan prediktabilitas. Selain itu, uji korelasi menunjukkan nilai rata-rata mendekati nol antara plainteks dan cipherteks, yang mengindikasikan rendahnya hubungan statistik dan tingginya ketahanan terhadap serangan kriptanalisis berbasis pola.

Temuan ini mendukung tujuan penelitian yang diuraikan pada bagian pendahuluan, yaitu menciptakan algoritma transposisi yang lebih aman dengan tingkat keacakan tinggi dan korelasi rendah antara data asli dan data terenkripsi. Algoritma ini berkontribusi secara langsung terhadap masalah keamanan data pada aplikasi yang memerlukan perlindungan kuat namun tetap efisien. Secara keseluruhan, penelitian ini memberikan solusi praktis dan teoritis

di bidang kriptografi, yang dapat menjadi referensi untuk pengembangan metode enkripsi yang lebih adaptif dan aman. Dari hasil penelitian menunjukkan bahwa penggunaan skema spiral dalam kotak transposisi ukuran 8 x 8 mampu memperkuat keamanan kriptografi dengan memutuskan pola-pola deterministik yang sering ditemukan dalam skema transposisi konvensional. Penerapan algoritma ini pada berbagai uji enkripsi juga menunjukkan tingkat korelasi yang rendah, yang merupakan indikator penting dalam menilai efektivitas sebuah algoritma kriptografi. Hasilnya, algoritma ini berhasil meningkatkan kerumitan dalam proses pemecahan pola, sehingga lebih sulit untuk dianalisis oleh kriptanalis.

REFERENSI

[1] R. Latifah, S. N. Ambo, dan S. I. Kurnia, "Modifikasi Algoritma Caesar Chiper Dan Rail Fence Untuk Peningkatan Keamanan Teks Alfanumerik Dan Karakter Khusus," 2017.

[2] A. Basuki, U. Paranita, dan R. Hidayat, "Perancangan Aplikasi Kriptografi Berlapis Menggunakan Algoritma Caesar, Transposisi, Vigenere, Dan Blok Chiper Berbasis Mobile," 2016.

[3] F. D. Paliama, "Perancangan Kriptografi Block Cipher Berbasis pada Teknik Formasi Permainan Bola,"2016.

[4] A. Widodo, "Perancangan Kriptografi Block Cipher Berbasis pada Teknik Bajak Sawah, Tanam Padi dan Panen Padi,"2015.

[5] K. D. Cahyono, "Perancangan Kriptografi Block Cipher dengan Langkah Permainan Engklek".

[6] M. A. Ineke Pekereng dan A. D. Wowor, "Square transposition: an approach to the transposition process in block cipher," 2021.

[7] H. Solís-Sánchez dan E. G. Barrantes, "Using the Logistic Coupled Map for Public Key Cryptography under a Distributed Dynamics Encryption Scheme," 2018.

[8] K. Paraditasari dan A. D. Wowor, "Desain Pembangkit Kunci Block Cipher Berbasis Csprng Chaos Menggunakan Fungsi Trigonometri,"2017.

[10] A. D. Wowor dan V. B. Liwandouw, "Domain Examination of Chaos Logistics Function As A Key Generator in Cryptography,"2018.

[11] I. Solihin dan A. P. U. Siahaan, "Implementasi Algoritma Super Playfair Chipher Dan Two Square Cipher Dalam Pengamanan Pesan Teks," 2017.

[12] R. Riyaldhi, Rojali, dan A. Kurniawan, "Improvement of Advanced Encryption Standard Algorithm With Shift Row and S.Box Modification Mapping in Mix Column," 2017.

[13] A. N. Setiawan, "lmplementasi Rancangan Algoritma Langkah Kuda (Permainan Catur) dan Anyaman Tali Sepatu Criss Cross Lacing dalam Kriptografi Block Cipher,"2015.

[14] W. M. Mauliku, "Perancangan dan Implementasi Algoritma Kriptografi Cipher BlockBerbasis pada Bentuk Piramida dan Linear Congruential Generator,"2015.

[15] A. K. Aziiz dan M. A. I. Pakereng, "Perancangan Teknik Kriptografi Block Cipher Berbasis Pola Batik Ceplok Yogyakarta," 2020.

[16] A. A. Lubis, N. P. Wong, I. Arfiandi, V. I. Damanik, dan A. Maulana, "Steganografi pada Citra dengan Metode MLSB dan Enkripsi Triple Transposition Vigenere Cipher,"2015.