JUSTIN
*Jurnal Sistem dan Teknologi Informasi*
p-ISSN : 2460-3562 / e-ISSN : 2620-8989

Vol. 8, No. 1, Januari 2020

# Perancangan Teknik Kriptografi Block Cipher Berbasis Pola Batik Ceplok Yogyakarta

Anriza Kurnia Aziiz [#1], Magdalena A. Ineke Pakereng[#2]

[#] *Progdi Teknik Informatika, Fakultas Teknologi Informasi, UKSW Salatiga*
*Jl. Diponegoro 52-60 Salatiga, Jawa Tengah, Indonesia*

[1]anrizakuriizz@gmail.com

[2]ineke.pakereng@uksw.edu

***Abstrak*— Kriptografi adalah suatu ilmu untuk menjaga keamanan dan kerahasiaan suatu informasi. Dalam penelitian ini dirancang Kriptografi *Block Cipher* 64 bit Berbasis Pola Batik Ceplok Yogyakarta guna membangun kriptografi baru. Dalam kritografi ini dirancang dengan 10 putaran, dimana setiap putaran terdapat 4 proses. Pada setiap putaran terdapat 4 pola untuk proses plaintext dan 4 pola untuk proses kunci. Di proses kedua dan keempat ditransformasikan dengan tabel S-BOX untuk mendapatkan *ciphertext* yang lebih acak. Pengujian juga dilakukan menggunakan *Avalanche Effect* dan nilai Korelasi dimana ratarata perubahan karakter mencapai 47,656%, sehingga dapat digunakan sebagai alternatif dalam mengamankan data.**

***Kata kunci*— Kriptografi, Block Cipher, S-BOX, Pola Batik Ceplok Yogykarta, Korelasi, Avalanche Effect**

## I. PENDAHULUAN

Enkripsi secara eksplisit dapat diartikan sebagai suatu proses untuk mengubah pesan (informasi) sehingga tidak dapat dilihat tanpa menggunakan kunci pembuka rahasia. Teknologi ini sudah digunakan sejak lama oleh kalangan militer dan intelejen. Saat ini, teknologi enkripsi dengan beberapa modifikasi sudah diaplikasikan untuk kepentingan umum, dalam aktivitas digital seperti merahasiakan data-data penting milik perorangan maupun perusahaan. Hasil statistik dari Breach Level Index (BLI) membuktikan, sepanjang 2016 telah terjadi 1.378.509.261 kehilangan atau pencurian data di seluruh dunia, atau sama dengan 3.776.738 data per hari, dan 157,364 per jam. Dari keseluruhan pelanggaran data di 2016 hanya 4 persen pembobolan data dianggap tidak berhasil karena data yang dicuri sudah terlebih dulu di enkripsi oleh perusahaan [1].

Maka dari itu, dapat dikatakan bahwa keamanan dalam proses pemindahan informasi sangat diperlukan. IT infrastruktur mulai gencar dalam merancang dan membangun untuk mengamankan informasi. Kriptografi hadir sebagai ilmu untuk menjaga kerahasiaan pesan/mengamankan informasi. Informasi yang dapat dibaca dan dipahami dengan bahasa tertentu diubah ke dalam bentuk sandi tertentu yang berstruktur huruf/kata/kalimat yang susah dipahami dari segi bahasa apapun. Salah satu algoritma nya adalah menggunakan algoritma Kriptografi *Block Cipher*. *Block Cipher* menggunakan kumpulan bit dengan panjang tetap yang disebut sebagai *block* dan kemudian dioperasikan dengan cipher kunci untuk nantinya ditransformasikan. Seiring kemajuan teknologi, makin banyak pula cara untuk memecahkan algoritma ini. Untuk itu salah satu cara untuk membuat data atau informasi menjadi lebih aman adalah dengan membuat pola atau algoritma baru untuk memodifikasi algoritma yang sudah ada.

Penelitian ini merupakan kriptografi *Block cipher* dengan menggunakan pendekatan pola batik ceplok Yogyakarta. Dari pola-pola tersebut akan dicari korelasi terbaik yang kemudian akan digunakan sebagai proses enkripsi dan dekripsi pesan *plaintext*. Beberapa motif-motif dalam batik ceplok Yogyakarta dijadikan pola pertukaran kode bit di dalamnya. Sehingga kemanan data menjadi lebih kuat dan data dapat digunakan sebagaimana mestinya.

## II. TINJAUAN PUSTAKA

Pada penelitian yang berjudul Perancangan Algoritma Transposisi dengan Nilai Indeks Berdasarkan Formasi Bola Basket pada *Block cipher*. Penelitian ini membahas mengenai perancangan algoritma *Block cipher* dengan formasi penyerangan bola basket untuk menghasilkan sebuah pola yang dapat digunakan untuk enkripsi dan dekripsi *plaintext* [2].

Submitted 19-10-2019; Revised 12-01-2020; Accepted 21-01-2020

Pada penelitian yang berjudul Perancangan Kriptografi *Block cipher* Berbasis Pada Anyaman Rambut Papua (ARAP). Penelitian ini membahas mengenai analisa korelasi dari pola anyaman rambut papua sehingga menghasilkan pola *block cipher* yang paling optimal untuk digunakan dalam proses enkripsi dan dekripsi [3].

Pada penelitian yang berjudul Penggunaan Pola Spiral untuk Perancangan P-Box dalam proses Transposisi pada *Block cipher* 128 Bit. Penelitian ini membahas perancangan kriptografi *block cipher* dengan menggunakan pola membuang batu kedalam air untuk membentuk gelombang yang menyerupai lingkaran yang berbasis pada *block cipher*. Kemudian dari pola yang terbentuk akan digunakan untuk proses enkripsi dan dekripsi *plaintext* [4].

Pada penelitian yang berjudul Analisis Permutasi Pola Transposisi dengan Game Sudoku dalam Rancangan Kriptografi *Block cipher*. Penelitian ini membahas perancangan pola kriptografi yang disesuaikan dengan teknik pada permainan sudoku 8x8 dimana semua angka dalam satu baris tidak boleh ada yang sama [5].

Pada penelitian yang berjudul Perancangan Kriptografi *Block cipher* Menggunakan Pola Kabel UTP *Straight* dan *Cross Over*. Penelitian ini membahas pola kriptografi yang akan digunakan dalam proses enkripsi dan dekripsi *plaintext*. Pola straight dijadikan dalam pertukaran kode bit pada *plaintext* sedangkan pola *cross over* digunakan pertukaran kode bit pada kunci [6].

Pada penelitian yang berjudul Perancangan Kriptografi Block Cipher 64 Bit Berbasis Pada Pola Formasi Sepak Bola 3-5-2. Algoritma yang digunakan dalam penelitian ini adalah algoritma block cipher 64 bit dengan menggunakan pola formasi sepak bola 3-5-2. Penggunaan formasi tersebut mempunyai tujuan agar kriptografi ini dapat menunjukkan ciri khas dari sebuah permainan Sepak Bola dalam sebuah team dan pola yang digunakan (Formasi 3-5-2) dipakai dalam proses pengambilan bit sehingga dapat menyembunyikan kerahasiaan data dengan lebih baik [7].

Pada penelitian yang berjudul Perancangan Kriptografi Block Cipher Berbasis Pola Formasi Futsal 1-2-1. Dalam penelitian ini membahas tentang sebuah metode kriptografi Block Cipher dengan rancangan formasi khusus dalam dunia futsal yang biasa disebut tiga satu dan dapat membuktikan bahwa pola ini dapat menyembunyikan kerahasiaan data dengan sangat baik [8].

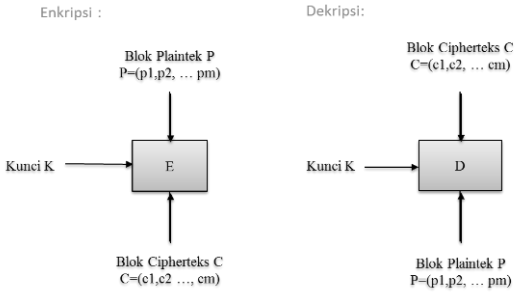
Pada penelitian yang berjudul Pengaruh S-Box Advance Encryption Standard (AES) pada Perubahan Ciphertext terhadap Perancangan Kriptografi Block Cipher 64 Bit Berbasis Pola Huruf U. Membahas mengenai algoritma kriptografi dengan pendekatan block cipher berbasis 64 bit dengan pola huruf U sebagai pola pengambilan bit-bit sebanyak 8 bit. Dimana juga terdapat pengaruh S-Box pada perubahan ciphertext. Pola huruf U digunakan karena dapat memenuhi bit-bit pada blok-blok yang ada yaitu 64 bit serta adanya transposisi pada pola huruf U [9].

Pada penelitian yang berjudul *Designing an algorithm with high Avalanche Effect*. Penelitian ini membahas tetang perancangan kriptografi block cipher berbasis 64 bit menggunakan gabungan kriptografi klasik dengan kirptogafi moderen untuk peningkatan *Avalanche Effect* [10].

Pada penelitian yang berjudul Perancangan Kriptografi Block Cipher Berbasis Pola Alat Musik Tifa Papua. Dalam penelitian ini, membahas mengenai kriptografi simetris yang mengenkripsi satu blok plaintext dengan jumlah bit tertentu dan menghasilkan blok ciphertext dengan jumlah bit yang sama, sedangkan pola yang digunakan (alat musik tifa Papua) dipakai dalam proses pengambilan bit [11].

Berdasarkan penelitian-penelitian sebelumnya terkait perancangan kriptografi *block cipher*, maka dilakukan penelitian tentang perancangan kriptografi *Block cipher* dengan memanfaatkan pola batik ceplok Yogyakarta. Dari pola-pola tersebut akan dicari korelasi terbaik yang kemudian akan digunakan sebagai proses enkripsi dan dekripsi pesan *plaintext*.

Skema proses enkripsi dan dekripsi *block cipher* secara umum digambarkan pada Gambar 1 [12].




Gambar. 1 Skema proses enkripsi dan dekripsi pada *block cipher*

Misalkan blok *plaintext* (P) yang berukuran m bit dinyatakan sebagai

$$P= (p1, p2, \dots pn) \quad (1)$$

Blok *ciphertext* (C) dinyatakan sebagai

$$C = (c1, c2, \dots cn) \quad (2)$$

Kunci (K) dinyatakan sebagai

$$K = (k1, k2, \dots kn) \quad (3)$$

Sehingga proses enkripsi adalah

$$EK(P) = C \quad (4)$$

Dan proses dekripsi adalah

$$DK(C) = P \quad (5)$$

Sebuah sistem kriptografi harus memenuhi lima-tupel (five-tuple) (P, C, K, E, D) dengan kondisi [13]:

1. P adalah himpunan berhingga dari *Plaintext*.
2. C adalah himpunan berhingga dari *Ciphertext*.
3. K merupakan ruang kunci (*keyspace*), adalah himpunan berhingga dari kunci.
4. Untuk setiap k $\in$K terdapat aturan enkripsi e_k$\in$ E dan berkorespodensi dengan aturan dekripsi d_k $\in$D. Setiap e_k:P$\rightarrow$C dan d_k:C$\rightarrow$P adalah fungsi sedemikian hingga d_k (e_k (x))=x untuk setiap *plaintext* x$\in$P.

Dalam pengujian menggunakan korelasi yang merupakan teknik statistik untuk mengukur kekuatan hubungan antar dua variabel dan untuk mengetahui bentuk hubungan antara dua variabel tersebut dengan hasil yang bersifat kuantitatif. Kekuatan hubungan antar dua variabel itu disebut dengan koefisien korelasi. Nilai koefisien akan selalu berada diantara -1 sampai +1. Untuk menentukan kuat atau lemahnya hubungan antara variabel yang diuji, dapat digunakan Tabel 1 [14].

TABEL I
UKURAN HURUF UNTUK TABEL

| Interval Koefisian | Tingkat Hubungan |
|---|---|
| 0,00 – 0,199 | Sangat Rendah |
| 0,20 – 0,399 | Rendah |
| 0,40 – 0,599 | Sedang |
| 0,60 – 0,799 | Kuat |
| 0,80 – 1,000 | Sangat Kuat |

Selain itu proses *Block cipher* ini menggunakan operasi XOR dimana output yang dihasilkan dari proses enkripsi akan susah ditebak, karena apabila kita melihat dasar dari XOR seperti berikut :

- 0 XOR 0 = 0
- 0 XOR 1 = 1
- 1 XOR 0 = 1
- 1 XOR 1 = 0

Maka apabila hasil output adalah 0 maka untuk mendapatkan input nya kita tidak tahu, bisa jadi input yang dihasilkan adalah 1 atau 0. Dasar tersebut digunakan untuk melakukan kriptografi *block cipher*.

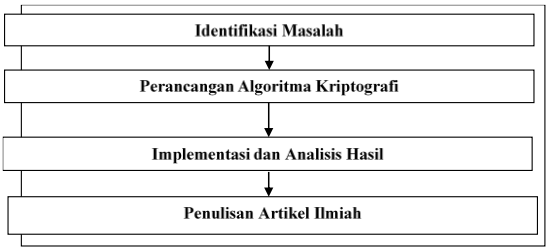
Kemudian S-BOX (Substitution Box) merupakan salah satu prinsip dalam perancangan blok cipher dimana proses s-box itu sendiri adalah mengganti karakter inputan dengan karakter yang sudah menjadi tetapan pada sebuah tabel. Secara teoritis, S-BOX adalah satu-satunya algoritma yang mempunyai kemampuan untuk membuat hubungan yang tidak linier antara *plaintext* dan *ciphertext*. Maka dari itu, penggunaan S-BOX ditujukan agar membuat kriptografi *block cipher* menjadi lebih acak. Hal ini dilakukan dengan cara mensubstitusikan bilangan *hexadecimal* ke dalam tabel S-BOX dan kemudian kita ambil output dari tabel S-BOX berupa bilangan *hexadecimal* yang baru.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 63 | 7C | 77 | 7B | F2 | 6B | 6F | C5 | 30 | 01 | 67 | 2B | FE | D7 | AB | 76 |
| 1 | CA | 82 | C9 | 7D | FA | 59 | 47 | F0 | AD | D4 | A2 | AF | 9C | A4 | 72 | C0 |
| 2 | B7 | FD | 93 | 26 | 36 | 3F | F7 | CC | 34 | A5 | E5 | F1 | 71 | D8 | 31 | 15 |
| 3 | 04 | C7 | 23 | C3 | 18 | 96 | 05 | 9A | 07 | 12 | 80 | E2 | EB | 27 | B2 | 75 |
| 4 | 09 | 83 | 2C | 1A | 1B | 6E | 5A | A0 | 52 | 3B | D6 | B3 | 29 | E3 | 2F | 84 |
| 5 | 53 | D1 | 00 | ED | 20 | FC | B1 | 5B | 6A | CB | BE | 39 | 4A | 4C | 58 | CF |
| 6 | D0 | EF | AA | FB | 43 | 4D | 33 | 85 | 45 | F9 | 02 | 7F | 50 | 3C | 9F | A8 |
| 7 | 51 | A3 | 40 | 8F | 92 | 9D | 38 | F5 | BC | B6 | DA | 21 | 10 | FF | F3 | D2 |
| 8 | CD | 0C | 13 | EC | 5F | 97 | 44 | 17 | C4 | A7 | 7E | 3D | 64 | 5D | 19 | 73 |
| 9 | 60 | 81 | 4F | DC | 22 | 2A | 90 | 88 | 46 | EE | B8 | 14 | DE | 5E | 0B | DB |
| A | E0 | 32 | 3A | 0A | 49 | 06 | 24 | 5C | C2 | D3 | AC | 62 | 91 | 95 | E4 | 79 |
| B | E7 | C8 | 37 | 6D | 8D | D5 | 4E | A9 | 6C | 56 | F4 | EA | 65 | 7A | AE | 08 |
| C | BA | 78 | 25 | 2E | 1C | A6 | B4 | C6 | E8 | DD | 74 | 1F | 4B | BD | 8B | 8A |
| D | 70 | 3E | B5 | 66 | 48 | 03 | F6 | 0E | 61 | 35 | 57 | B9 | 86 | C1 | 1D | 9E |
| E | E1 | F8 | 98 | 11 | 69 | D9 | 8E | 94 | 9B | 1E | 87 | E9 | CE | 55 | 28 | DF |
| F | 8C | A1 | 89 | 0D | BF | E6 | 42 | 68 | 41 | 99 | 2D | 0F | B0 | 54 | BB | 16 |

Gambar. 2 Tabel S-BOX

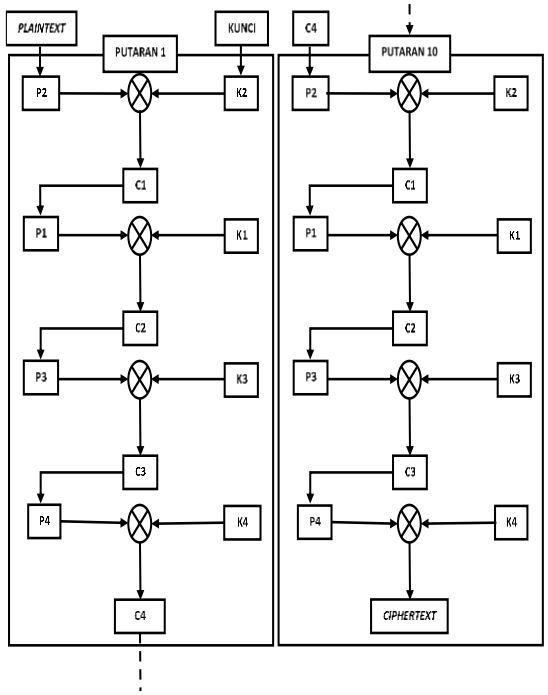
## III. METODE PENELITIAN DAN PERANCANGAN ALGORITMA

Secara umum penelitian terbagi ke dalam 4 (empat) tahapan, yaitu: (1) tahap identifikasi masalah, (2) tahap perancangan, (3) tahap implementasi dan analisis hasil, (4) tahap penulisan artikel ilmiah.




Gambar. 3 Tahapan penelitian

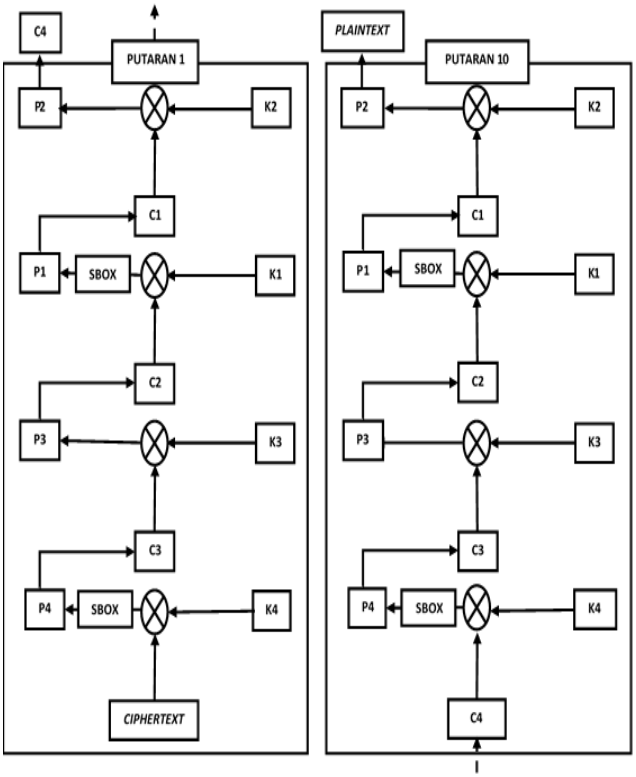
Tahapan Penelitian pada Gambar 3 dapat dijelaskan sebagai berikut; (1) Tahap Identifikasi Masalah: Pada tahapan ini dilakukan analisis terhadap permasalahan yang ada, terkait dengan proses perancangan Kriptografi *Block cipher* Berbasis Pola Batik Ceplok Yogyakarta. (2) Tahap Perancangan Kriptografi: Pada tahap ini akan dilakukan perancangan Kriptografi *Block cipher* Berbasis Pola Batik Ceplok Yogyakarta dengan menggunakan 4 pola yang telah dibuat dan menggunakan Tabel S-BOX sebagai tambahan agar terbentuk *ciphertext* yang lebih acak. Untuk pembuatan kunci, proses enkripsi dan proses dekripsi dikombinasikan dengan XOR. (3) Tahap Pengujian Kriptografi: Pada tahap ini dilakukan pengujian terhadap kriptografi yang telah dibuat. Pengujian proses 24 kombinasi yang akan menghasilkan nilai korelasi terendah, kemudian dilakukan proses enkripsi dan dekripsi, yang terakhir yaitu mencari *avalanche effect* terbesar. (4) Tahap Penulisan Artikel Ilmiah: Dalam tahap terakhir ini dilakukan penulisan artikel mengenai proses Perancangan Kriptografi *Block cipher* berbasis Pola Batik Ceplok Yogyakarta.




Gambar. 4 Alur proses enkripsi

Gambar 4 merupakan alur proses enkripsi. Konsep dari proses enkripsi dapat dijabarkan sebagai berikut: a) Menyiapkan *plaintext*; b) Mengubah *plaintext* menjadi biner sesuai dalam tabel ASCII; c) Dalam proses enkripsi, *plaintext* dan kunci akan melewati empat proses pada setiap putaran, yaitu : 1) Putaran pertama *Plaintext* 1 (P1) melakukan transformasi dengan pola batik ceplok dan di XOR dengan Kunci 1 (K1) menghasilkan *Ciphertext* 1 (C1); 2) *Plaintext* 2 (P2) melakukan transformasi dengan pola batik ceplok dan kemudian *Plaintext* 2 (P2) dilakukan proses S-BOX untuk menghasilkan bilangan biner yang baru, lalu di XOR dengan Kunci 2 (K2) menghasilkan *Ciphertext* 2 (C2), dan tahapan tersebut akan berlanjut sampai proses keempat kemudian *Plaintext* 4 (P4) dilakukan proses S-BOX untuk menghasilkan bilangan biner yang baru, kemudian proses dilanjutkan sehingga menghasilkan *Ciphertext* 4 (C4) ; 3) *Ciphertext* 4 (4) masuk pada putaran kedua dengan alur proses yang sama dengan putaran pertama, dan tahapan tersebut akan berlanjut sampai putaran ke-10 yang menghasilkan *Ciphertext* Akhir.

Gambar 5 menunjukkan alur proses dekripsi. Konsep proses dekripsi tersebut dijelaskan sebagai berikut: a) Menyiapkan *ciphertext* dan kunci; b) Mengubah *ciphertext* dan kunci menjadi biner sesuai dalam tabel ASCII; c) dalam perancangan dekripsi, *ciphertext* dan kunci akan melewati empat proses pada setiap putaran; d) Proses pertama *Ciphertext* (C) diproses dengan pola dan di XOR dengan Kunci 4 (K4) dari putaran 10, menghasilkan P4; d) P4 tersebut kemudian dilakukan proses S-BOX sehingga menghasilkan biner yang baru yang kemudian menjadi C3 di putaran 10; e) Masuk pada proses dua, C3 diproses dengan pola dan di XOR dengan Kunci 3 (K3) dari putaran 10, menghasilkan P3. P3 akan menjadi C2 pada putaran selanjutnya; f) Masuk pada proses ketiga, C2 di XOR dengan Kunci 1 (K1) menghasilkan P1 yang kemudian dilakukan proses S-BOX agar menghasilkan bilangan biner yang baru. Bilangan biner yang baru tersebut dimasukkan ke *Plaintext* 1 (P1). P1 akan menjadi C1 pada proses selanjutnya; g) Masuk ke proses keempat, P1 di XOR kan dengan Kunci 2 (K2) menghasilkan *plaintext* akhir yaitu P1.



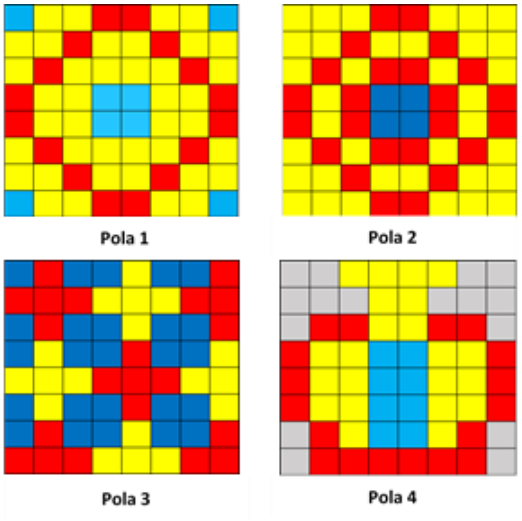

Gambar. 5 Alur proses dekripsi

## IV. HASIL DAN PEMBAHASAN

Bagian ini membahas tentang algoritma perancangan kriptografi *block cipher* berbasis pola batik ceplok Yogyakarta secara lebih rinci. Dalam algoritma ini pola yang terdapat pada contoh batik ceplok Yogyakarta digunakan sebagai proses pemasukan dan pengambilan bit. Pola tersebut ditunjukkan pada Gambar 5.

Pada Gambar 6 menunjukkan empat pola yang berbeda, dimana pola-pola tersebut menunjukkan pola-pola yang terdapat pada batik ceplok Yogyakarta. Berdasarkan pola-pola yang sudah dirancang, dilakukan pengujian korelasi dengan mengkombinasikan urutan pola untuk menemukan nilai korelasi terbaik. Pengujian dilakukan menggunakan contoh *plaintext* "DIESUKSW" menggunakan kunci "BUDAYAKU".

Berdasarkan hasil pengujian korelasi, maka hasil terkecil yang akan digunakan sebagai acuan perancangan dalam proses enkripsi dan dekripsi.




Gambar. 6 Pola Batik Ceplok Yogyakarta

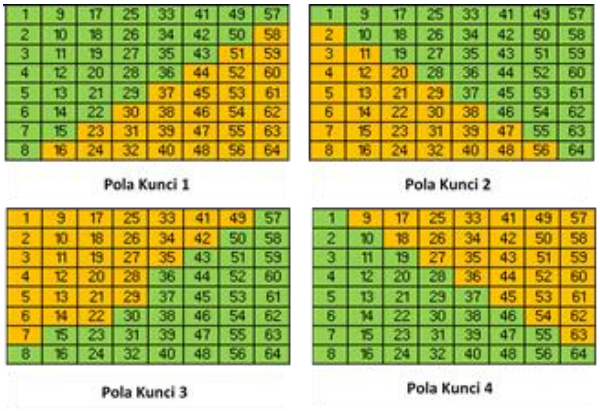
TABEL III
HASIL KORELASI SETIAP KOMBINASI POLA BATIK CEPLOK YOGYAKARTA

| Pola | Rata-Rata | Pola | Rata-Rata |
|---|---|---|---|
| 1-2-3-4 | 0,203443777 | 3-1-2-4 | 0,167138614 |
| 1-2-4-3 | 0,309212605 | 3-1-4-2 | 0,068999878 |
| 1-3-2-4 | 0,655646347 | 3-2-1-4 | 0,094572274 |
| 1-3-4-2 | 0,340229681 | 3-2-4-1 | 0,032662108 |
| 1-4-2-3 | 0,442350815 | 3-4-1-2 | 0,099859446 |
| 1-4-3-2 | 0,403175215 | 3-4-2-1 | 0,38101642 |
| 2-1-3-4 | 0,016724212 | 4-1-2-3 | 0,145015858 |
| 2-1-4-3 | 0,089202026 | 4-1-3-2 | 0,336560574 |
| 2-3-1-4 | 0,426891967 | 4-2-1-3 | 0,439582513 |
| 2-3-4-1 | 0,240791721 | 4-2-3-1 | 0,155924 |
| 2-4-1-3 | 0,157898994 | 4-3-1-2 | 0,030187913 |
| 2-4-3-1 | 0,039772404 | 4-3-2-1 | 0,12270925 |

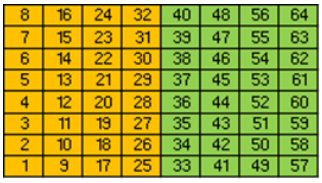
Tabel 2 menunjukkan hasil kombinasi pola dan mendapatkan nilai korelasi terbaik pada kombinasi pola 2-1-3-4. Kombinasi ini lah yang akan digunkan untuk melanjutkan proses enkripsi hingga putaran ke-10 untuk menghasilkan *ciphertext*.

Telah dijelaskan bahwa perancangan kriptografi ini dilakukan sebanyak 10 putaran, dan disetiap putaran memiliki 4 proses untuk mendapatkan hasil akhir yaitu *ciphertext*. Proses pertama *plaintext* dan kunci diubah kedalam bentuk ASCII kemudian diubah lagi kedalam biner. Kemudian bit-bit *plaintext* diproses dengan pola pemasukan dan pengambilan kedalam kolom matriks 8x8 menggunakan bagian dari pola batik yang berbeda-beda pada setiap proses. Kemudian di setiap proses dilakukan X-OR dari *Plaintext* (P) dan kunci (K) menghasilkan *ciphertext* (C) sampai proses keempat di setiap putaran. Kemudian diulang terus sampai putaran ke-10 dan hingga menghasilkan *Ciphertext* akhir.

Untuk menjelaskan secara detail proses pemasukan bit dalam matriks maka diambil proses 1 pada putaran 1 sebagai contoh. Misalkan angka 1 merupakan inisialisasi setiap bit yang merupakan hasil konversi *plaintext* maka urutan bit adalah sebagai berikut 1, 2, 3, 4, …..64.

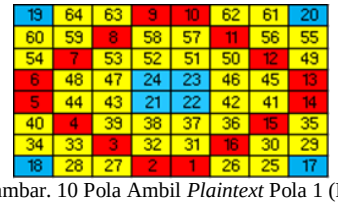



Gambar. 7 Pola Pemasukan Kunci



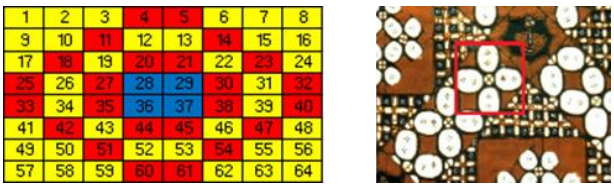
Gambar. 8 Pola Ambil Semua Kunci

Gambar. 9 Pola Pemasukan *Plaintext* dari Pola 1 (P1)

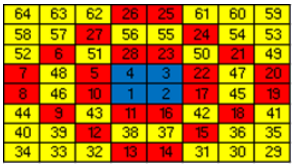


Gambar. 10 Pola Ambil *Plaintext* Pola 1 (P1)

Gambar 7 merupakan pola yang digunakan untuk memasukkan kunci kedalam tabel 8x8 dengan urutan sesuai angka pada pola Gambar 7. Gambar 8 digunakan untuk mengambil kunci berupa bit-bit angka dari Gambar 7. Gambar 9 merupakan pola masuk dari pola 1 (P1) yang digunakan untuk memasukkan setiap 8 bit dari karakter *plaintext*. Gambar 10 digunakan untuk mengambil *plaintext* berupa bit-bit angka dari Gambar 9.

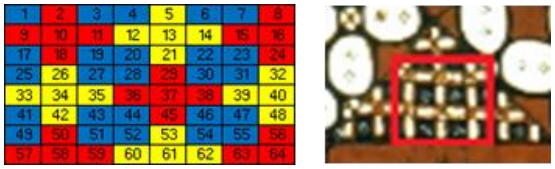


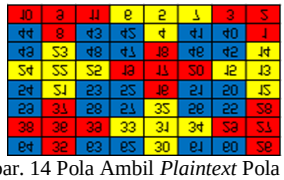
Gambar. 11 Pola Pemasukan *Plaintext* dari Pola 2 (P2)



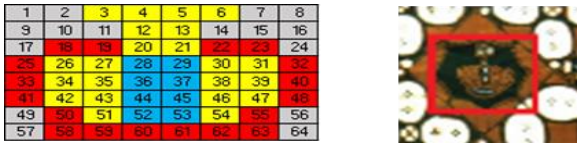
Gambar. 12 Pola Ambil *Plaintext* Pola 2 (P2)

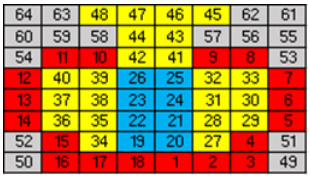
Gambar 11 merupakan pola masuk dari pola 2 (P2) yang digunakan untuk memasukkan setiap 8 bit dari karakter *plaintext*. Gambar 12 digunakan untuk mengambil *plaintext* berupa bit-bit angka dari Gambar 11.


Gambar. 13 Pola Pemasukan *Plaintext* dari Pola 3 (P3)


Gambar. 14 Pola Ambil *Plaintext* Pola 3 (P3)

Gambar 13 merupakan pola masuk dari pola 3 (P3) yang digunakan untuk memasukkan setiap 8 bit dari karakter *plaintext*. Gambar 14 digunakan untuk mengambil *plaintext* berupa bit-bit angka dari Gambar 14.


Gambar. 15 Pola Pemasukan *Plaintext* dari Pola 4 (P4)



Gambar. 16 Pola Ambil *Plaintext* Pola 4 (P4)

Gambar 15 merupakan pola masuk dari pola 4 (P4) yang digunakan untuk memasukkan setiap 8 bit dari karakter *plaintext*. Gambar 16 digunakan untuk mengambil *plaintext* berupa bit-bit angka dari Gambar 15.

Dengan menggunakan pola-pola yang sudah ditetapkan, dilakukan proses enkripsi dan dekripsi yang dilakukan sebanyak 10 putaran dimana setiap putaran terdapat 4 proses. Proses enkripsi dan dekripsi dijelaskan lebih lanjut pada Tabel 3 berupa algoritma dan *pseudocode* yang menjelaskan lebih detail mengenai algoritma enkripsi dan dekripsi.

TABEL IIIII
ALGORITMA ENKRIPSI DAN DEKRIPSI

| No. | Proses Enkripsi | No. | Proses Dekripsi |
|---|---|---|---|
| 1. | Masukkan plaintext | 1. | Masukkan ciphertext |
| 2. | Plaintext diubah ke DECIMAL | 2. | Ciphertext diubah ke DECIMAL |
| 3. | DECIMAL diubah ke BINER | 3. | DECIMAL diubah ke BINER |
| 4. | Bit BINER dimasukkan ke kolom matriks 8x8 P2 dengan pola pemasukan plaintext | 4. | Bit BINER dimasukkan ke kolom matriks 8x8 C4 dengan pola pemasukan plaintext |
| 5. | Bit pada kolom matrik diambil menggunakan pola pengambilan pola 2 | 5. | C4 di-XOR dengan K4 menghasilkan P4 |
| 6. | Bit pengambilan dimasukkan lagi kedalam matrik mendapatkan hasil akhir P2 | 6. | P4 diproses dengan pola pemasukan plaintext |
| 7. | P2 dilakukan proses S-BOX. | 7. | P4 dilakukan proses S-BOX |
| 8. | P2 di-XOR dengan K2 menghasilkan C1 | 8. | Hasil proses P4 dimasukkan kedalam matriks 8x8 lagi dengan pola pengambilan pola 4 |
| 9. | C1 menjadi P1 untuk proses selanjutnya | 9. | P4 menjadi C3 untuk proses selanjutnya |
| 10. | Bit pada kolom matrik diambil menggunakan pola pengambilan pola 1 | 10. | C3 di-XOR dengan K3 menghasilkan P3 |
| 11. | Bit pengambilan dimasukkan lagi kedalam matrik mendapatkan hasil akhir P1 | 11. | P3 diproses dengan pola pemasukan plaintext |
| 12. | P1 di-XOR dengan K1 menghasilkan C2 | 12. | Hasil proses P3 dimasukkan kedalam matriks 8x8 lagi dengan pola pengambilan pola 3 |
| 13. | C2 menjadi P3 untuk proses selanjutnya | 13. | P3 menjadi C2 untuk proses selanjutnya |
| 14. | Bit pada kolom matrik diambil menggunakan pola pengambilan pola 3 | 14. | C2 di-XOR dengan K1 menghasilkan P1 |
| 15. | Bit pengambilan dimasukkan lagi kedalam matrik mendapatkan hasil akhir P3 | 15. | P1 diproses dengan pola pemasukan plaintext |
| 16. | P3 di-XOR dengan K3 menghasilkan C3 | 16. | Hasil proses P1 dimasukkan kedalam matriks 8x8 lagi dengan pola pengambilan pola 1 |
| 17. | C3 menjadi P4 untuk proses selanjutnya | 17. | P1 menjadi C1 untuk proses selanjutnya |
| 18. | Bit pada kolom matrik diambil menggunakan pola pengambilan pola 4 | 18. | C1 di-XOR dengan K2 menghasilkan P2 |
| 19. | Bit pengambilan dimasukkan lagi | 19. | P2 diproses dengan pola pemasukan *plaintext* |

| No. | Proses Enkripsi | No. | Proses Dekripsi |
|---|---|---|---|
| | kedalam matrik mendapatkan hasil akhir P4 | | |
| 20. | P4 dilakukan proses S-BOX | 20. | P2 dilakukan proses S-BOX |
| 21. | P4 di-XOR dengan K4 menghasilkan C4 | 21. | Hasil proses P2 dimasukkan kedalam matriks 8x8 lagi dengan pola pengambilan pola 2 |
| 22. | C4 diubah ke DECIMAL | 22. | P2 diubah ke DECIMAL |
| 23. | DECIMAL diubah ke CHAR untuk mendapatkan *Ciphertext* akhir. | 23. | DECIMAL diubah ke CHAR untuk mendapatkan *Plaintext* awal. |

Dari Tabel 3 dapat kita amati bahwa proses enkripsi menghasilkan *Ciphertext* akhir, dan proses dekripsi menghasilkan *Plaintext* awal. Kemudia algoritma proses kunci, dijelaskan sebagai berikut:

1. Masukkan Kunci
2. Kunci diubah ke DECIMAL
3. DECIMAL ke BINER
4. Bit BINER dimasukkan ke kolom K2 dengan pola pemasukan Kunci
5. Bit kunci diambil dengan pola pengambilan Kunci
6. BINER hasil pengambilan dimasukkan kedalam kolom matrik K2
7. K2 = K1
8. K1 dimasukkan ke kolom matrik K1 dengan pola pemasukan
9. Bit kunci diambil dengan pola pengambilan Kunci
10. BINER hasil pengambilan dimasukkan kedalam kolom matrik K1
11. K1 = K3
12. K3 dimasukkan ke kolom matrik K3 dengan pola pemasukan
13. Bit kunci diambil dengan pola pengambilan Kunci
14. BINER hasil pengambilan dimasukkan kedalam kolom matrik K3
15. K3 = K4
16. K4 dimasukkan ke kolom matrik K4 dengan pola pemasukan
17. Bit kunci diambil dengan pola pengambilan Kunci
18. BINER hasil pengambilan dimasukkan kedalam kolom matrik K4

Pseudocode proses enkripsi dan dekripsi dijelaskan pada Tabel 4 berikut.

TABEL IV
*PSEUDOCODE* PROSES ENKRIPSI DAN DEKRIPSI

**Proses Enkripsi**
{Program ini digunakan untuk melakukan proses enkripsi data 64 *bit*}

Kamus
P,K,P1,K1,P2,K2,P3,K3,P4,K4, = integer
C,C1,C2,C3,C4 = integer

```
Start
  C1 <- P2 ⊕ K2
    Input P
    Read P
    P to ASCII
      ASCII to BINER
      Dari BINER = blok matriks P, masukkan BINER
      P menggunakan Pola pemasukan awal
        Dari blok matriks P = BINER, ambil bit
        P dengan Pola Batik Ceplok Yogyakarta
        Plaintext 2 = blok matriks P2
    Output P1
    Input K
    Read K
      K to ASCII
      ASCII to BINER
      Dari BINER = blok matriks K, masukkan BINER
      K menggunakan Pola pemasukan awal
        Dari blok matriks K = BINER, ambil bit
        K dengan Pola Kunci 2 = blok matriks K2
    Output K2
  Print C1
  C2 <- P1 ⊕ K1
    Input P
    Read P
    P to ASCII
      ASCII to BINER
      Dari BINER = blok matriks P, masukkan BINER
      P menggunakan Pola pemasukan awal
        Dari blok matriks P = BINER, ambil bit
        P dengan Pola Batik Ceplok Yogyakarta
        Plaintext 1
        BINER to HEXA
        Dari HEXA = Tabel S-Box, masukkan HEXA
        HEXA substitusi menggunakan S-Box
        HEXA S-Box to BINER = block matriks P1
    Output P1
    Input K
    Read K
      K to ASCII
      ASCII to BINER
      Dari BINER = blok matriks K, masukkan BINER
      K menggunakan Pola pemasukan awal
        Dari blok matriks K = BINER, ambil bit
        K dengan Pola Kunci 1 = blok
```

```
Start
    P4 <- C4 ⊕ K4
      Input C4
      Read C4
        C4 to ASCII
        ASCII to BINER
        Dari BINER = blok matrik C4, masukan BINER
      Output C4
      Input K
      Read K
        K to ASCII
        ASCII to BINER
        Dari BINER = blok matriks K, masukkan BINER
        K menggunakan Pola pemasukan awal
          Dari blok matriks K = BINER, ambil bit
          K dengan Pola Kunci 4 = blok matriks K4
      Output K4
      C4 ⊕ K4
    Output P4
      Dari kolom matrik P4 = BINER, ambil bit P4
      BINER to HEXA
        Dari HEXA = Tabel S-Box, masukan HEXA
        HEXA ditranformasi menggunakan S-Box
        Dari BINER P4 = kolom matrik P4, masukan
        BINER menggunakan pola pengambilan 4
    Print P4
    P3 <- C3 ⊕ K3
      Input C3
      Read C3
        C3 to ASCII
        ASCII to BINER
        Dari BINER = blok matrik C3, masukan BINER
      Input K
      Read K
        K to ASCII
        ASCII to BINER
        Dari BINER = blok matriks K, masukkan BINER
        K menggunakan Pola pemasukan awal
          Dari blok matriks K = BINER, ambil bit
          K dengan Pola Kunci 3 = blok matriks K3
      Output K3
    C3 ⊕ K3
    Print P3
    P1 <- C2 ⊕ K1
      Input C2
```

```
                matriks K1
        Output K1
    Print C2
    C3 <- P3 ⊕ K3
        Input P
        Read P
        P to ASCII
            ASCII to BINER
            Dari BINER = blok
matriks P, masukkan BINER
            P menggunakan Pola
pemasukan awal
                Dari blok matriks P
=  BINER, ambil bit
                P dengan Pola Batik
                Ceplok Yogyakarta
                Plaintext 3  = blok
                matriks P3
        Output P3
        Input K
        Read K
            K to ASCII
            ASCII to BINER
            Dari BINER = blok
matriks K, masukkan BINER
            K menggunakan Pola
pemasukan awal
                Dari blok matriks K
=  BINER, ambil bit
                K dengan Pola
                Kunci 3  = blok
                matriks K3
        Output K3
    Print C3
    C4 <- P4 ⊕ K4
        Input P
        Read P
        P to ASCII
            ASCII to BINER
            Dari BINER = blok
matriks P, masukkan BINER
            P menggunakan Pola
pemasukan awal
                Dari blok matriks P
=  BINER, ambil bit
                P dengan Pola Batik
                Ceplok Yogyakarta
                Plaintext 4
                BINER to HEXA
                Dari HEXA = Tabel
                S-Box, masukkan
                HEXA
                HEXA substitusi
                menggunakan S-
                Box
                HEXA S-Box to
                BINER = block
                matriks P4
        Output P4
        Input K
        Read K
            K to ASCII
            ASCII to BINER
            Dari BINER = blok
matriks K, masukkan BINER
            K menggunakan Pola
pemasukan awal
                Dari blok matriks K
=  BINER, ambil bit
                K dengan Pola
                Kunci 4 = blok
                matriks K1
            Read C2
                C2 to ASCII
                ASCII to BINER
                Dari BINER = blok
matrik C2, masukan BINER
            Input K
            Read K
                K to ASCII

                ASCII to BINER
                Dari BINER = blok
matriks K, masukkan BINER
                K menggunakan Pola
pemasukan awal
                    Dari blok matriks
                    K =  BINER,
                    ambil bit
                    K dengan Pola
                    Kunci 1  = blok
                    matriks K1
                Output K1
            C2 ⊕ K1
            Output P1
                Dari kolom matrik P1 =
BINER, ambil bit P1
                BINER to HEXA
                    Dari HEXA = Tabel
                    S-Box, masukan
                    HEXA
                    HEXA ditranformasi
                    menggunakan S-Box
                    Dari BINER P1 =
                    kolom matrik P1,
                    masukan
                    BINER
                    menggunakan pola
                    pengambilan 1
            Print P1
            P2 <- C1 ⊕ K2
                Input C1
                Read C1
                    C1 to ASCII
                    ASCII to BINER
                    Dari BINER = blok
matrik C1, masukan BINER
                Input K
                Read K
                    K to ASCII

                    ASCII to BINER
                    Dari BINER = blok
matriks K, masukkan BINER
                    K menggunakan Pola
pemasukan awal
                        Dari blok matriks
                        K =  BINER,
                        ambil bit
                        K dengan Pola
                        Kunci 2  = blok
                        matriks K2
                    Output K2
                C1 ⊕ K2
                Print P2
Repeat
End
```

```
        Output K4
    Print C4
Repeat
End
```

Tabel 5 merupakan hasil dari proses S-BOX yang dilakukan pada setiap putaran untuk proses *Plaintext* 2 dan *Plaintext* 4. Proses S-BOX dilakukan agar *Ciphertext* yang dihasilkan pada setiap akhir putaran menjadi lebih acak.

TABEL V
HASIL PERUBAHAN P2 DAN P4 SETIAP PUTARAN SETELAH DILAKUKAN PROSES S-BOX

| **Putaran** | ***Plaintext*** | **Hexa Sebelum Proses S-BOX** | **Hexa Sesudah Proses S-BOX** |
|---|---|---|---|
| **1** | P2 | 980F791F61FD9D31 | E2FBAFCBD821752E |
| | P4 | 1924A116A105A178 | 8EA6F1FFF136F1C1 |
| **2** | P2 | A54ACE02534BC35E | 295CEC6A50CC339D |
| | P4 | 0400F0E1AE1C3D92 | 305217E0BEC48B74 |
| **3** | P2 | 42C0005BCD903F89 | F61F5257809625F2 |
| | P4 | D3E50A68F909D7C4 | A92AA3F769400D88 |
| **4** | P2 | 7E6EF976518B6209 | 8A45690F70CEAB40 |
| | P4 | 24182CA17D829DA4 | A63442F11311751D |
| **5** | P2 | 9B6DDF5B0853A2FC | E8B3EF57BF501A55 |
| | P4 | 14B0C23F416AA2A3 | 9BFCA825F8581A71 |
| **6** | P2 | 37D5832E31FAC981 | B2B541C32E141291 |
| | P4 | 47F1228E0D488A97 | 162B94E6F3D4CF85 |
| **7** | P2 | 8BCA42149D025208 | CE10F69B756A48BF |
| | P4 | BA108AD113EFEDDB | C07CCF518261539F |
| **8** | P2 | 5A3DA79ECA47ABA4 | 468B89DF10160E1D |
| | P4 | A5AD8B0AD10E0DB3 | 2918CEA351D7F34B |
| **9** | P2 | 45FB577EB45D236C | 6863DA8AC68D32B8 |
| | P4 | 1549E04521DC5455 | 2FA4A0687B93FDED |
| **10** | P2 | B5E15B621C439253 | D2E057ABC4647450 |
| | P4 | 7A6D308623C8E5AD | BDB308DC32B12A18 |

Hasil dari proses enkripsi di setiap putaran (dengan urutan 1-10) adalah *ciphertext* (C) yang berupa char. Hasil enkripsi di putaran ke 10 adalah *ciphertext* akhir, seperti ditunjukkan pada Tabel 6.

TABEL VI
HASIL *CIPHERTEXT* SETIAP PUTARAN PADA PROSES ENKRIPSI

| Putaran | Hasil Hexadesimal | Hasil *Char* |
|---|---|---|
| 1 | CCF3B5BEA877BA94 | Ìóµ¾¨wº” |
| 2 | 720753A1E785C021 | rS¡ç…A! |
| 3 | EB7FE7B6300146DD | ë • ç¶0FÝ |
| 4 | E4616B04A503E48 | äa°JP>H |
| 5 | D9A9EC64A1195124 | Ù©ìd¡Q$ |
| 6 | 547ED0A7AA9584D0 | T~Ð§ª•„Ð |
| 7 | 82298B10DB2018CA | ‚)‹Û Ê |
| 8 | 6B4D8AE2896B81E | kMŠâ–¸- |
| 9 | 6DF1E42922D2B6B8 | mña)"Ò¶¸ |
| 10 | FFE64C9D6BF0614D | ÿæ L • kðaM |

Hasil dari proses dekripsi di setiap putaran (dengan urutan 10-1) adalah *plaintext* (P) yang berupa char. Hasil dekripsi di putaran ke 1 adalah *plaintext* akhir, seperti ditunjukkan pada Tabel 7.

TABEL VII
HASIL *CIPHERTEXT* SETIAP PUTARAN PADA PROSES DEKRIPSI

| Putaran | Hasil Hexadesimal | Hasil *Char* |
|---|---|---|
| 1 | 44494553554B5357 | DIESUKSW |
| 2 | CCF3B5BEA877BA94 | Ìóµ¾¨wº” |
| 3 | 720753A1E785C021 | rS¡ç…À! |
| 4 | EB7FE7B6300146DD | ë • ç¶0FÝ |
| 5 | E4616B04A503E48 | äa°JP>H |
| 6 | D9A9EC64A1195124 | Ù©ìd¡Q$ |
| 7 | 547ED0A7AA9584D0 | T~Ð§ª•„Ð |
| 8 | 82298B10DB2018CA | ‚)‹Û Ê |
| 9 | 6B4D8AE2896B81E | kMŠâ–¸- |
| 10 | 6DF1E42922D2B6B8 | mña)"Ò¶¸ |

Nilai korelasi antara *plaintext* dan *ciphertext* dapat digunakan untuk mengukur seberapa acak hasil enkripsi (*ciphertext*) dengan *plaintext*. Nilai korelasi sendiri berkisar dari 0 sampai 1, dimana jika nilai kolerasi mendekati 1 maka *plaintext* dan *ciphertext* memiliki nilai yang sangat berhubung, tetapi jika mendekati 0 maka *plaintext* dan *ciphertext* tidak memiliki nilai yang berhubungan.

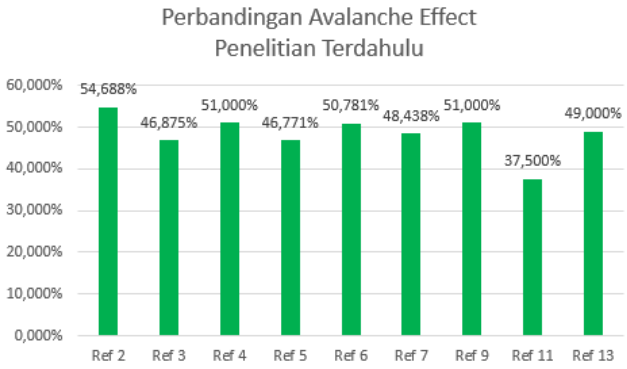
TABEL VIII
NILAI KORELASI SETIAP PUTARAN

| Putaran | Nilai Korelasi (absolut) |
|---|---|
| 1 | 0,350640613 |
| 2 | 0,224026201 |
| 3 | 0,271143398 |
| 4 | 0,199146369 |
| 5 | 0,076104989 |
| 6 | 0,120354667 |
| 7 | 0,042539768 |
| 8 | 0,125496836 |
| 9 | 0,4978222 |
| 10 | 0,39830648 |

Tabel 8 menunjukkan nilai korelasi antara *plaintext* awal dengan hasil *plaintext* di setiap putaran, dan dapat disimpulkan bahwa algoritma kriptografi *block cipher* berbasis pola batik ceplok Yogyakarta memiliki nilai korelasi lemah dan menghasilkan nilai korelasi yang acak.

Salah satu karakteristik untuk menentukan berhasil atau tidaknya suatu algoritma kriptografi adalah dengan melihat *avalanche effect*-nya. Pengujian dilakukan dengan merubah karakter yang terdapat pada *plaintext* awal, sehingga akan menghasilkan perbedaan pada setiap putarannya. Suatu algoritma kriptografi dikatakan baik jika perubahan bit yang dihasilkan berkisar antara 45-60% (sekitar separuhnya, 50% adalah hasil yang sangat baik). Hal ini dikarenakan perubahan tersebut berarti membuat perbedaan yang cukup sulit untuk kriptanalis melakukan serangan [15].

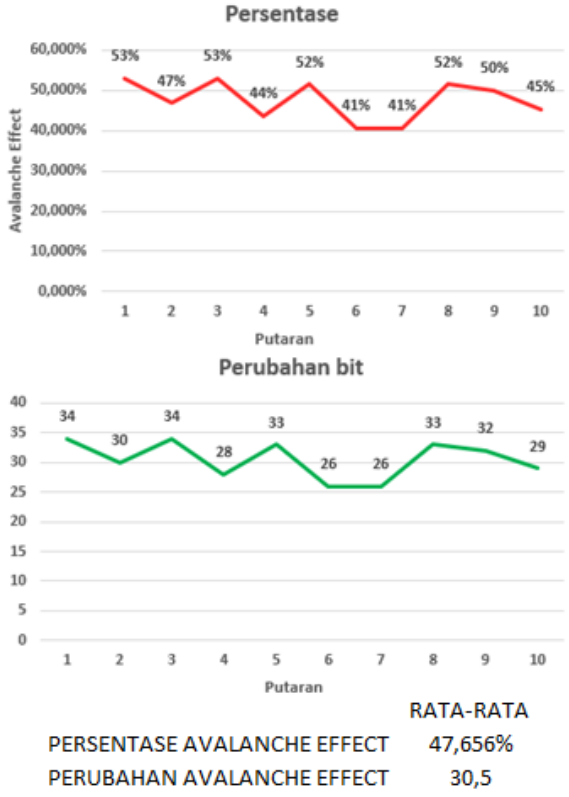
### *A. Pengujian*




Gambar. 17 Grafik Perbandingan *Avalanche Effect* Penelitian Terdahulu

Gambar 17 merupakan grafik yang menggambarkan hasil rata-rata nilai *Avalanche Effect* dari penelitian terdahulu. Dari grafik tersebut dapat dilihat bahwa penelitian dengan Referensi 11 mempunyai nilai rata-rata 37,500% yang berarti penelitian tersebut tidak termasuk kedalam kategori algoritma yang baik karena diluar dari *range* 45-60%. Sedangkan penelitian lainnya termasuk kedalam algoritma kriptografi yang baik karena termasuk kedalam *range* 45-60%. Bahkan nilai *avalanche effect* untuk Referensi 4, 6, 9, dan 13 termasuk sangat baik karena mendekati angka 50% yang berarti termasuk kategori sangat baik [15].

Gambar 18 adalah hasil dari pengujian *Avalanche Effect* dari Perancangan Teknik Kriptografi Block Cipher Berbasis Pola Batik Ceplok Yogyakarta, pada kasus ini *plaintext* awal adalah “DIESUKSW” yang kemudian diubah menjadi “Anriza21”. Terjadi perubahan bit pada setiap putarannya, pada putaran ke-1 dan ke-3 perubahan bitnya terjadi cukup besar yaitu 53,125% dengan arti pada putaran ini terjadi perubahan bit yang baik, tetapi juga terjadi perubahan bit yang kecil pada putaran ke-6 dan ke-7 yaitu sebesar 40,625% ini berarti perubahan bitnya kurang baik. Berdasarkan hasil putaran pertama sampai dengan putaran ke sepuluh dapat disimpulkan bahwa rata-rata hasil pengujian *Avalanche Effect* ini yaitu sebesar 47,656% yang berarti termasuk kategori yang baik [15].




Gambar. 18 Grafik *Avalanche Effect*

## V. KESIMPULAN

Berdasarkan penelitian yang dilakukan, dapat disimpulkan bahwa kriptografi *block cipher* 64 bit berbasis pola batik ceplok Yogyakarta dapat dikatakan sebagai sistem kriptografi. Dalam proses enkripsi, rancangan kriptografi *block cipher* berbasis pola batik ceplok Yogyakarta ini menghasilkan output yang acak sehingga dapat digunakan sebagai alternatif dalam pengamanan data. Dalam pengujian *avalanche effect* yang dilakukan pun menunjukkan bahwa proses enkripsi di setiap putaran memiliki rata-rata perubahan yang mencapai 47,656% yang berarti algoritma kriptografi ini berhasil dan termasuk ke dalam kategori yang baik. Walaupun termasuk ke dalam kategori yang baik, penelitian ini masih kurang baik apabila dibandingkan dengan penelitian terdahulu yang kebanyakan mempunyai nilai rata-rata *avalanche effect* lebih mendekati angka 50% yang berarti algoritma kriptografinya termasuk sangat baik.

## REFERENSI

[1] Berita Satu, "Teknologi Enkripsi, Solusi Terbaik Pengamanan Data," [Online]. Available: http://www.beritasatu.com/iptek/426799-teknologi-enkripsi-solusi-terbaik-pengamanan-data.html. [Accessed 29 November 2018].

[2] Priyoko, " Perancangan Algoritma Transposisi dengan Nilai Indeks Berdasarkan Formasi Bola Basket pada *Block cipher*," 2016. [Online]. Available: http://repository.uksw.edu/handle/123456789/13563. [Accessed 18 10 2018].

[3] S. Mamoba, "Perancangan Kriptografi *Block cipher* Berbasis Pada Anyaman Rambut Papua (ARAP)," 19 June 2018. [Online]. Available: http://repository.uksw.edu/handle/123456789/11290. [Accessed 18 10 2018].

[4] N. D. Ledewara, "Penggunaan Pola Spiral untuk Perancangan P-Box dalam proses Transposisi pada *Block cipher* 128 Bit," 2016. [Online]. Available: http://repository.uksw.edu/handle/123456789/13579. [Accessed 19 10 2018].

[5] D. P. Mahendra, "Analisis Permutasi Pola Transposisi dengan Game Sudoku dalam Rancangan Kriptografi *Block cipher*," 2016. [Online]. Available: http://repository.uksw.edu/handle/123456789/13520. [Accessed 18 10 2018].

[6] S. Atiq, "Perancangan Kriptografi *Block cipher* Menggunakan Pola Kabel UTP Straight dan Cross Over," 2016. [Online]. Available: http://repository.uksw.edu/handle/123456789/13568. [Accessed 18 10 2018].

[7] K. Tryanto, "Perancangan Kriptografi Block Cipher 64 Bit Berbasis Pada Pola Formasi Sepak Bola 3-5-2" 2017. [Online]. Available: https://repository.uksw.edu/handle/123456789/13500. [Accessed 19 10 2018].

[8] Louhenapessy, N.M., Pakereng, M. A. I., 2017, Perancangan Kriptografi Block Cipher Berbasis Pola Formasi Futsal 1-2-1. Salatiga : Jurusan Teknik Informatika Universitas Kristen Satya Wacana.

[9] Parapat, F. A. C., Pakereng, M. A. I., 2017, Pengaruh S-Box pada Perubahan Ciphertext Terhadap Perancangan Kriptografi Block Cipher 64 Bit Berbasis Pola Huruf U. Salatiga : Jurusan Teknik Informatika Universitas Kristen Satya Wacana.

[10] Ramanujam, S., Karuppiah, M. 2011. Designing an algorithm with high Avalanche Effect. IJCSNS International Journal.

[11] Heipon, Y. A., Pakereng, M. A. I., 2017, Perancangan Kriptografi Block Cipher Berbasis Pola Alat Musik Tifa Papua. Salatiga : Jurusan Teknik Informatika Universitas Kristen Satya Wacana.

[12] M. R, Kriptografi, Bandung: Informatika, 2006.

[13] A. J. Leodrian, "Pengaruh Perubahan *Ciphertext* terhadap Perancangan Kriptografi *Block cipher* 64 Bit Berbasis Pola Ikatan Jimbe dengan Menggunakan Kombinasi S-Box," 2016. [Online]. Available: http://repository.uksw.edu/handle/123456789/13488. [Accessed 20 10 2018].

[14] Sugiyono, Metode Penelitian Bisnis: Pendekatan Kuantitatif, Kualitatif, Kombinasi, dan R&D, Bandung: Alfabeta, 2017.

[15] Sugiyanto., "Pengembangan Algoritma Advanced Encryption Standard pada Sistem Keamanan SMS Berbasis Android Menggunakan Algoritma Vigenere," *ULTIMATICS*, vol. VIII, no. 02, pp. 135-137, 2016.