**BULLETIN OF INFORMATION TECHNOLOGY (BIT)**
Vol 3, No 2, Juni 2022, Hal 149 -154
ISSN 2722-0524 (media online)
DOI 10.47065/ bit.v3i1.302
https://journal.fkpt.org/index.php/BIT

# Pembangkitan Kunci Beaufort Cipher Dengan Teknik Blum-blum Shub pada Pengamanan Citra Digital

**Eferoni Ndruru, Taronisokhi Zebua**
Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Budidarma , Medan, Indonesia
Email: [1]ronindruru@gmail.com, [2]taronizeb@gmail.com
Email Penulis Korespondensi: *taronizeb@gmail.com

**Abstrak**−Keamanan data tidak hanya karena kompleksitas algoritma yang digunakan, tetapi salah satu faktor terpenting adalah keamanan kunci yang digunakan. Salah satu data paling populer yang perlu diamankan saat ini adalah citra digital. Namun, akhir-akhir ini, algoritma yang digunakan untuk mengamankan data sering diserang. Hal ini dikarenakan struktur formasi kunci yang digunakan masih lemah. Salah satu algoritma keamanan data yang membentuk kunci sederhana adalah beaufort cipher. Kunci yang digunakan pada algoritma ini merupakan rangkaian karakter yang saling Hal ini bergantung pada duplikasi karakter kunci untuk membentuk kunci untuk setiap elemen yang dilindungi. Pembangkitan kunci berdasarkan blum-blum shub dapat menghasilkan kunci yang sangat acak dan sulit ditebak, sehingga kekuatan kunci yang digunakan dapat meningkatkan ketahanan data yang dilindungi. Kunci yang dihasilkan berdasarkan blum-blum shub nantinya akan digunakan dalam proses enkripsi dan dekripsi nilai piksel citra digital.

**Kata Kunci:** Beaufort Cipher; Keamanan Data; Blum-blum Shub; Pembangkit Kunci; Citra

**Abstract**−Data security is not only due to the complexity of the algorithms used, but one of the most important factors is the security of the keys used. One of the most popular data that needs to be backed up today is digital images. However, these days, the algorithms used to back up data are often attacked. This is because the structure of the key formations used is still weak. One of the data security algorithms that form a simple key is the Beaufort cipher. It relies on the duplication of key characters to form the key for each protected element. Key generators based on blum-blum shub technology can generate very random and difficult-to-guess keys, so the strength of the keys you use can enhance the resilience of protected data. Keys generated based on blum-blum shub technology will later be used in the process of encrypting and decrypting pixel values in digital images.

**Keywords :** Beaufort Cipher; Data Security; Blum-blum Shub; Key Generate; Image

## 1. PENDAHULUAN

Pemanfaatan citra digital sebagai salah satu media dalam berkomunikasi saat ini telah banyak digunakan. Citra digital yang didistribusikan dapat saja memiliki nilai penting, bersifat pribadi atau rahasia. Namun perkembangan teknologi saat ini sangat berdampak terhadap lahirnya permasalahan penyalahgunaan data yang didistribusikan. Oleh karena itu, maka sangat diperlukan teknik pengamanan yang dapat diterapkan pada citra digital.

Kriptografi merupakan salah satu ilmu yang yang berperan penting dalam bidang pengamanan informasi. Kriptografi memiliki teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi misalnya kerahasiaan dan integritas data, serta otentikasi. Kuat lemahnya metode kriptografi tidak terletak dari hasil enkripsi atau ciphertext, melainkan terletak pada kunci yang digunakan, oleh sebab itu kunci merupakan jantung dari pertahanan data tersebut agar tidak dapat diakses atau dipecahkan oleh orang-orang yang tidak bertanggung jawab[1][2]. Algoritma di dalam kriptografi terbagi menjadi dua, yaitu algoritma kunci simetri dan algoritma kunci asimetri. Salah satu yang termasuk dalam algoritma kunci simetri adalah beafort cipher[3].
Jumlah karakter kunci yang digunakan dalam proses enkripsi maupun dekripsi berdasarkan beaufort cipher ini harus sama dengan jumlah plain, sehingga setiap karakter plain harus tepat berpasangan dengan satu kunci masing-masing. Jumlah karakter kunci yang harus berbanding lurus dengan jumlah karakter plaintext dapat menyulitkan pengguna untuk menghafal kata kunci yang digunakan[4]. Oleh karena itu, algoritma ini memperkenankan untuk menggunakan kata kunci yang lebih pendek dan melakukan pengulangan kata kunci tersebut hingga jumlahnya sama dengan jumlah karakter plaintext. Hal inilah menjadi salah satu kelemahan dari algoritma ini.

Salah satu cara yang dapat dilakukan untuk mengatasi kelemahan beaufort cipher di atas adalah dengan melakukan pembangkitan kunci yang lebih acak. Kunci yang dibangkitkan secara acak tahan terhadap serangan analisa frekuensi tinggi[5]. Salah satu algorima yang dapat digunakan untuk membangkitkan kunci nilai-nilai adalah pembangkit kunci blum blum shub.

Pembangkit kunci blum blum shub merupakan salah satu pembangkit kunci yang cara kerjanya sederhana, namun paling bagus dan kompleks dalam mengamankan data. Pembangkit kunci ini aman dalam mengamankan data, karena pemecahan pembangkit kunci ini setara dengan memecahkan metode quadratic residue problem yang pada gilirannya akan memecahkan number pseucode lengkap yang merupakan basis kriptografi dan menjadikan pembangkit kunci blum blum shub ini menjadi salah satu jenis pembangkit kunci yang paling disukai khususnya dalam proses pembangkitan kunci, karena kunci yang dihasilkan susah ditebak[6][7].

BULLETIN OF INFORMATION TECHNOLOGY (BIT)
Vol 3, No 2, Juni 2022, Hal 149 -154
ISSN 2722-0524 (media online)
DOI 10.47065/ bit.v3i1.302
https://journal.fkpt.org/index.php/BIT

Penelitian ini menguraikan bagaimana prosedur yang dilakukan untuk membangkitkan kunci yang digunakan pada beaufort cipher untuk mengenkripsi dan dekripsi citra digital berformat jpg. Proses pembangkitan kunci dilakukan berdasarkan pembangkit kunci blum blum shub, artinya kunci yang digunakan pada proses enkripsi dan dekripsi adalah kunci yang dibangkitkan berdasarkan pembangkit kunci blum blum shub, sehingga diharapkan keamanan citra digital yang diamankan dapat lebih terjamin dan algoritma yang digunakan untuk mengamankannya tidak mudah dipecahkan oleh pihak yang tidak bertanggung jawab.

# 2. METODOLOGI PENELITIAN

## 2.1 Kriptografi

Kriptografi merupakan salah satu teknik yang mempelajari agar pesan yang disitribusikan kepada orang lain dapat sampaikan dengan aman[7]. Implementasi teknik kriptografi dapat menjaga kerahasiaan inforasi yang terkandung di dalam sebuah data, sehingga orang lain tidak dapat mengetahuinya. Beberapa hal yang harus dicapai dalam mengimplementasikan kriptografi adalah kerahasiaan (*confidentiality*), integritas data (data *integrity*), autentikasi (*authentication*), ketiadaan penyangkalan (*nonrepudiation*) [8][2]. Teknik kriptografi mengamankan data dengan proses pengubahan data asli menjadi simbol-simbol yang tidak dapat dimengerti lagi. Hal ini dilakukan berdasarkan algoritma-algoritma kriptografi yang telah ada. Ada tiga fungsi utama dari teknik kriptografi[9], yaitu enkripsi, dekripsi, kunci.

Beberapa kekuatan yang dimiliki oleh algoritma kriptografi dalam proses mengenkripsi data [10], yaitu:

1. Konfusi/pembingungan (*confusion*), yaitu suatu proses dimana teks sulit dikembalikan pada bentuk awal secara tanpa melalui proses dekripsi.
2. Difusi/peleburan *(difusion*), yaitu suatu proses dimana karakteristik suatu teks dihilangkan sehingga mengamankan suatu informasi..

## 2.2 Beaufort Cipher

Algoritma ini termasuk dalam ke dalam kelompok kriptografi klasik dimana kunci (K) pada *beaufort cipher* adalah urutan karakter-karakter $K = k1 \ldots k_d$. Nilai $k1$ merupakan nilai atau jumlah pergeseran dari alfabet ke-$i$ [4]. Sehingga dapat diartikan bahwa jumlah karakter kunci yang digunakan berbanding lurus dengan jumlah karakter plain yang ingin diamankan, sehingga masing-masing karakter plain harus memiliki pasangan kunci. Hal ini yang menyebabkan algoritma ini hampir sama dengan algoritma *vigeneere cipher*. Adapun formulasi yang digunakan dalam proses enkripsi dan dekripsi [4][11], adalah :

Formula proses enkripsi :

$Ci = Ek(Mi) = (Ki - Mi) \text{ Mod } 26$ .....(1)

Formula proses dekripsi :

$Mi = Dk(Ci) = (Ki - Mi) \text{ Mod } 26$ .....(2)

Dimana :

Mi = Data yang diamankan
Ci = Cipher Ki = Kunci
Ek = Fungsi Enkripsi Dk = Fungsi Dekripsi

Agar seluruh nilai ASCI dapat diakses, maka nilai modulus dapat diganti dengan 256.

## 2.3 Pembangkit Kunci Blum-blum Shub

Pembangkit Kunci *blum blum shub* merupakan pembangkit kunci bilangan acak yang sangat sederhana dan paling bagus secara kompleksitas teoritis[12][13]. Pembangkit kunci *blum blum shub* (BBS) dibuat pada tahun 1986 oleh Lenore Blum, Manuel Blum, Michael Shub yang dirancang dengan dasar teori bilangan. BBS memiliki bentuk persamaan[12][14], yaitu :

$X_{n+1} = X_n^2 \text{ Mod } m$……..(3)

dimana :

X = barisan *bit* acak yang dihasilkan
n = bilangan bulat prima *blum*
m = hasil perakalian p dan q

Adapun langkah-langkah penggunaan pembangkit kunci BBS yang telah ditentukan[12][14], yaitu :

1. Pilih dua bilangan prima rahasia p dan q, yang masing-masing kongruen dengan 3 mod 4 dalam hal ini bilangan prima yang digunakan adalah bilangan prima yang bernilai besar dimana semakin besar bilangan prima yang dipilih maka semakin tinggi tingkat keamanannya. Berikut adalah persamaannya :
   $p \equiv 3 \text{ mod } 4$ dan $q \equiv 3 \text{ mod } 4$

BULLETIN OF INFORMATION TECHNOLOGY (BIT)
Vol 3, No 2, Juni 2022, Hal 149 -154
ISSN 2722-0524 (media online)
DOI 10.47065/ bit.v3i1.302
https://journal.fkpt.org/index.php/BIT

2. Kemudian persamaan di atas, kalikan p dan q yang disebut bilangan bulat *blum* dengan persamaan :
   $n = p \times q$……..(4)
3. Pilih bilangan bulat acak lain yang dilambangkan dengan s, sebagai umpan sedemikian. Berikut adalah persamaannya :
   ≤ s < n dimana s adalah relatif bilangan prima
4. Barisan bilangan acak didapatkan dengan cara melakukan sepanjang yang diinginkan. Adapun langkah-langkah barisan bilangan acak, yaitu :
   Hitung $x_i$ dengan persamaan :
   $x_i = x_{i-1}^2 \bmod n$…….(5)
   Kemudian hasilkan $z_i$ yang merupakan bit-bit dari $x_i$. Bit yang diambil bisa merupakan *least significant bit* atau hanya satu *bit* atau sebanyak j *bit* dan bilangan *bit* acak dapat digunakan langsung atau diformat dengan aturan tertentu hingga mejadi bilangan bulat
5. Barisan *bit* acak adalah $z_1, z_2, z_3, \ldots, z_i$

Adapun kelebihan dari pembangkit kunci *blum blum shub* yaitu secara statistik memiliki sifat-sifat yang bagus atau lulus uji keacakan, tahan terhadap serangan yang bertujuan untuk memprediksi bilangan acak yang dihasilkan dan yang menjadi kelemahan dari pembangkit kunci *blum blum shub* yaitu jika hasil bilangan prima yang dipilih kecil, maka penyadap dapat mengetahui dengan cepat barisan *bit* yang dibangkitkan[14]

### 2.4 Citra Digital

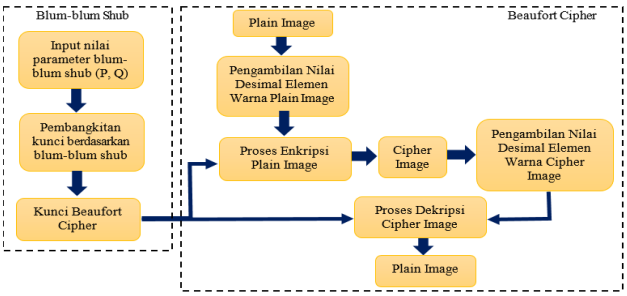
Salah satu representatif dari citra yang diambil oleh mesin dengan bentuk pendekatan berdasarkan sampling dan kuantisasi disebut dengan citra digital[15]. Besarnya nilai tingkat kecerahan yang dinyatakan dalam nilai tingkat keabuan (*grayscale*) sesuai dengan jumlah bit biner yang digunakan oleh mesin merupakan sampling pada citra. Sebuah citra digital juga dapat dikelompokan menjadi dua jenis yaitu gambar bitmap dan gambar vektor berdasarkan tingkat kuantisasi warnanya.
Gambar *bitmap* adalah gambar yang terbentuk dari *pixel*, dengan sikap *pixel*-nya mempunyai warna tertentu. Umumnya foto dan gambar menggunakan format gambar *bitmap* yang biasanya diperoleh dari proses *scanner*, camera digital, video *capture*, contohnya filenya bmf,gif,jpg. Gambar vektor merupakan gambar yang dihasilkan dari perhitungan matematis dan tidak berdasarkan *pixel*[15].

## 3. HASIL DAN PEMBAHASAN

### 3.1 Analisa Pembangkitan Kunci Beaufort Cipher berdasarkan Teknik Blum-blum Shub

*Beaufort cipher* bekerja seperti algoritma *vegenere cipher* untuk mengamankan data rahasia atau data penting, yaitu dengan melakukan penggantian setiap karakter data yang diamankan, sehingga dihasilkan karakter baru yang menjadi sandi dari data asli. *Ciphertext* didapatkan dengan melakukan proses penambahan nilai decimal dari karakter data asli dengan nilai decimal kunci. Jumlah karakter kunci yang digunakan harus sama dengan jumlah karakter data asli. Bila jumlah karakter kunci tidak sama dengan jumlah karakter data asli, maka dilakukan proses pengulangan data kunci hingga jumlahnya sama dengan jumlah karakter data asli. Salah satu kelemahan *beaufort cipher* adalah penggunaan kunci yang berulang, sehingga bila salah satu potongan kata kunci berhasil ditemukan oleh penyerang, maka kata kunci keseluruhan yang digunakan untuk mengenkripsi data asli dapat diketahui dengan mudah.

Penelitian ini menjelaskan bagaimana mengoptimalkan kunci yang digunakan dalam proses enkripsi dan dekripsi. Optimasi yang dilakukan adalah bagaimana menghasilkan karakter kunci yang acak berdasarkan metode blum-blum shub. Hal ini dilakukan agar menghasilkan kunci enkripsi dan dekripsi yang jauh lebih rumit dan tidak mudah diketahui oleh penyerang. Berikut ini adalah skema prosedur pembangkitan kunci *beaufort cipher* berdasarkan teknik blum-blum sub untuk mengamankan citra digital.




**Gambar 1**. Skema Pembangkitan Kunci *Beaufort Cipher* Berdasarkan *Blum-Blum Shub*

BULLETIN OF INFORMATION TECHNOLOGY (BIT)
Vol 3, No 2, Juni 2022, Hal 149 -154
ISSN 2722-0524 (media online)
DOI 10.47065/ bit.v3i1.302
https://journal.fkpt.org/index.php/BIT

### 3.1.1 Pembangkitan Kunci Berdasarkan Teknik Blum-Blum Shub

Sebagai contoh citra yang dijadikan sebagai dalam penelitian adalah citra berwarna dan berformat jpg dengan resolusi 3x3 *pixel*. Bila dikonversi menjadi nilai warna, maka dapat diketahui bahwa setiap *pixel* terdiri dari 3 elemen warna (*red*, *green* dan *blue*). Berdasarkan dimenasi citra sampel, maka dihasilkan jumlah keseluruhan nilai yang akan dienkripsi adalah 27 nilai (3 x 3 *pixel* x 3 elemen warna). Nilai-nilai decimal dari setiap elemen warna ditunjukkan pada tabel 1.

**Tabel 1**. Nilai Desimal Elemen Warna Citra Sampel

| Pixel 0 | | | Pixel 1 | | | Pixel 2 | | |
|---|---|---|---|---|---|---|---|---|
| R | G | B | R | G | B | R | G | B |
| 200 | 170 | 75 | 165 | 80 | 170 | 230 | 55 | 103 |
| 45 | 187 | 87 | 37 | 135 | 158 | 199 | 165 | 221 |
| 126 | 251 | 46 | 125 | 85 | 135 | 78 | 100 | 85 |

Nilai-nilai warna citra pada tabel 1 di atas akan dienkripsi berdasarkan *beaufort cipher* dan tentunya harus memiliki 27 kunci yang dalam penelitian ini akan dibangkitkan berdasarkan *blum-blum shub* seperti proses di bawah ini :

1. Proses pembangkitan kunci berdasarkan pembangkit BBS diawali dengan pemilihan 2 bilangan prima rahasia dan dianggap sebagai nilai p dan q dengan syarat bila p atau 2 mod 3 = 3. Misalnya dipilih p = 59 dan q = 71.
2. Nilai p dan 2 dikalikan untuk menghasilkan nilai n, maka 59 x 71 = 4189
3. Memilih nilai s dengan syarat s dan n relatif prima, sehingga didapat s = 21.
   Nilai $x_0$ dihitung dengan $s^2$ mod n = $21^2$ mod 4189 = 441 mod 4189 = 441
4. Menentukan nilai LSB (8 bit) untuk membentuk karakter kunci
   Penentuan nilai LSB dilakukan dengan melanjutkan proses pencarian nilai $x_i$ syarat nilai LSB yang diambil adalah hasil dari nilai $x_i$ mod 2. Proses untuk mendapakan hasil nilai mod $x_2$ dan seterusnya berpatokan pada hasil $x_1$ atau hasil $x_i$ sebelumnya.
   Misalnya untuk mengambil 8 bit LSB untuk membentuk karakter kunci pertama, maka dilakukan proses pembangkitan nilai x hingga 8 kali.
   $x_1 = x_0^2$ mod n = $441^2$ mod 4189 = 441 mod 4189 = 441; LSB = 1
   $x_2 = x_1^2$ mod n = $441^2$ mod 4189 = 194481 mod 4189 = 1787; LSB=1
   $x_3 = x_2^2$ mod n = $1787^2$ mod 4189 = 3193369 mod 4189 = 1351; LSB=1
   $x_4 = x_3^2$ mod n = $1351^2$ mod 4189 = 1825201 mod 4189 = 2986; LSB=0
   $x_5 = x_4^2$ mod n = $2986^2$ mod 4189 = 8916196 mod 4189 = 2004; LSB=0
   $x_6 = x_5^2$ mod n = $2004^2$ mod 4189 = 4016016 mod 4189 = 2954; LSB=1
   $x_7 = x_6^2$ mod n = $2954^2$ mod 4189 = 8726116 mod 4189 = 429; LSB=1
   $x_8 = x_7^2$ mod n = $429^2$ mod 4189 = 184041 mod 4189 = 3914; LSB=0
   sehingga gabungan LSB yang didapatkan dari $x_1$-$x_8$ adalah 11100110 atau dalam nilai desimal adalah 230 dan akan menjadi nilai kunci untuk elemen warna red dari *pixel* ke-0

   Proses pembentukan kunci ke-2 dilakukan seperti di atas, dengan nilai x yang selalu berpatokan pada nilai akhir dari x sebelumnya (nilai x akhir adalah 3914.
   $x_1 = x_0^2$ mod n = $3914^2$ mod 4189 = 15319396 mod 4189 = 223; LSB=1
   $x_2 = x_1^2$ mod n = $223^2$ mod 4189 = 49729 mod 4189 = 3650; LSB=0
   $x_3 = x_2^2$ mod n = $3650^2$ mod 4189 = 13322500 mod 4189 = 1480; LSB=0
   $x_4 = x_3^2$ mod n = $1480^2$ mod 4189 = 2190400 mod 4189 = 3742; LSB=0
   $x_5 = x_4^2$ mod n = $3742^2$ mod 4189 = 14002564 mod 4189 = 2926; LSB=0
   $x_6 = x_5^2$ mod n = $2926^2$ mod 4189 = 8561476 mod 4189 = 3349; LSB=1
   $x_7 = x_6^2$ mod n = $3349^2$ mod 4189 = 11215801 mod 4189 = 1848; LSB=0
   $x_8 = x_7^2$ mod n = $1848^2$ mod 4189 = 3415104 mod 4189 = 1069; LSB=0
   sehingga hasil LSB dari $x_9$-$x_{16}$ adalah 10000100 atau dalam desimal adalah bernilai 132. Nilai ini akan dijadikan kunci untuk *pixel*-0 elemen warna *green*.

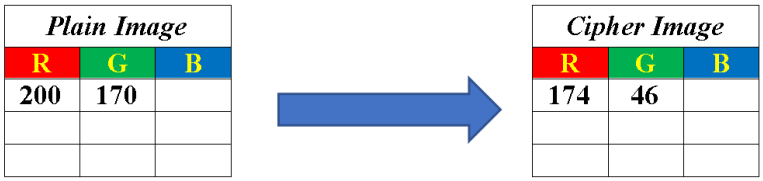
Proses pembangkitan 27 buah kunci untuk mengenkripsi dan dekripsi citra digital, dilakukan dengan cara yang sama seperti di atas.

### 3.1.2. Proses Enkripsi Berdasarkan Beaufort Cipher

Prosedur enkripsi berdasarkan *beaufort cipher* dilakukan dengan menambahkan nilai pesan yang dienkripsi (dalam hal ini nilai elemen warna citra) dengan nilai pasangan kunci yang digunakan dan dimoduluskan dengan 256.
Contoh :

BULLETIN OF INFORMATION TECHNOLOGY (BIT)
Vol 3, No 2, Juni 2022, Hal 149 -154
ISSN 2722-0524 (media online)
DOI 10.47065/ bit.v3i1.302
https://journal.fkpt.org/index.php/BIT

Nilai *pixel*-0 elemen warna *red* dari citra sampel adalah 200 dan nilai kunci ke-0 yang didapatkan berdasarkan BBS adalah 230, sehingga :
*Cipher pixel*-1 elemen warna *red* = (200+230) mod 256 = 174
*Cipher pixel*-2 elemen warna *green* = (170+132) mod 256 = 46
Proses ini dilakukan hingga seluruh elemen warna dari masing-masing citra terenkripsi

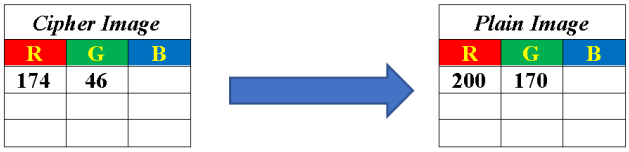



**Gambar 2.** Representasi Nilai *Cipher Image* Hasil Proses Enkripsi

### 3.1.3. Proses Dekripsi Berdasarkan Beaufort Cipher

Proses dekripsi tidak jauh berbeda dengan konsep proses enkripsi yang diawali dengan proses pembangkitan bilangan acak yang dijadikan sebagai kunci. Proses pembangkitan kunci dilakukan dengan menggunakan parameter nilai p, q dan s yang sama seperti yang digunakan pada proses enkripsi, kemudian akan dibangkitkan bilangan acak sebanyak jumlah elemen warna citra yang didekripsi. Berdasarkan proses enkripsi, maka nilai paramter p, q, s yang digunakan adalah p = 59, q = 71 dan s =21, sehingga akan menghasilkan nilai-nilai kunci yang sama seperti hasil pembangkitan kunci pada proses enkripsi.

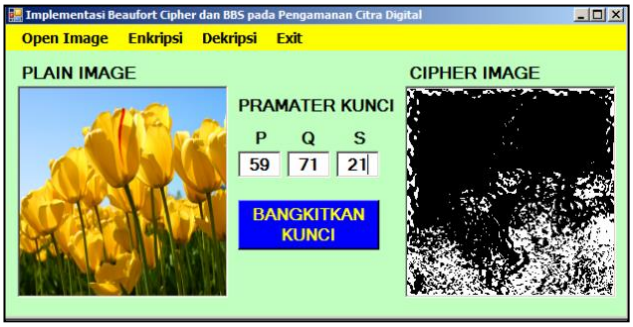
Formulasi proses dekripsi dilakukan dengan mengurangkan nilai setiap elemen warna cipher image dengan nilai kunci pasangannya kemudian dimoduluskan dengan 256, sehingga didapatkan citra asli (*plain image*).
Contoh :
Kunci yang dihasilkan untuk elemen warna ke-1 *pixel* ke-1 = 230
Kunci yang dihasilkan untuk elemen warna ke-2 *pixel* ke-1 = 132
Nilai elemen warna ke-1 *pixel* ke-1 *cipher image* = 174
Nilai elemen warna ke-2 *pixel* ke-1 *cipher image* = 46
sehingga proses dekripsi sebagai berikut :
*plain image pixel*-1 elemen-1 = (174 - 230) mod 256 = 200
*Plain image pixel*-2 elemen-2 = (46 - 132) mod 256 = 170
Proses ini dilakukan hingga seluruh elemen warna dari masing-masing *cipher image* dikembalikan menjadi nilai semula (*plain image*).




**Gambar 3.** Representasi Nilai *Plain Image* Hasil Proses Dekripsi

## 3.2 Implementasi

Aplikasi yang dibangun untuk mengimplementasikan proses pembangkitan kunci dan proses enkripsi serta dekripsi, hanya terdiri dari satu form yang dapat digunakan untuk melakukan proses pembangkitan kunci, proses enkripsi dan dekripsi citra yang diamankan.




**Gambar 4.** *Form* Pembangkitan Kunci, Enkripsi dan Dekripsi

BULLETIN OF INFORMATION TECHNOLOGY (BIT)
Vol 3, No 2, Juni 2022, Hal 149 -154
ISSN 2722-0524 (media online)
DOI 10.47065/ bit.v3i1.302
https://journal.fkpt.org/index.php/BIT

Sebelum melakukan proses enkripsi atau dekripsi, maka pengguna harus menginput citra yang akan dienkripsi (*plain image*) ataupun *cipher image*, kemudian menginput nilai parameter kunci yang digunakan (nilai p, q dan s). Sebelum melakukan proses enkripsi maupun dekripsi, maka pengguna harus melakukan proses pembangkitan kunci terlebih dahulu berdasar kan nilai-nilai parameter kunci yang telah diinput sebelumnya. Berdasarkan proses ini, maka akan dihasilkan deret acak kunci yang dapat digunakan untuk melakukan proses enkripsi maupun dekripsi

## 4. KESIMPULAN

Berdasarkan pembahasan yang dilakukan, dapat disimpulkan bahwa kunci yang dibangkitkan berdasarkan teknik pembangkitan blum-blum shub dapat menghasilkan kunci yang sangat acak dan tidak menyebabkan kunci yang berulang. Keacakan kunci ini dapat mengoptimalkan algoritma beuafort cipher untuk menghasilkan cipher image yang sangat sulit dikenali pola aslinya oleh para penyerang, sehingga dapat menjamin keamanannya.

## UCAPAN TERIMAKASIH

Terima kasih disampaikan kepada pihak Kampus Universitas Budidarma Medan dan LPPM Universitas Budidarma Medan.

## REFERENCES

[1] E. N. T. Guruh M Arindra Pratama, “No Title,” Metod. Catalan Number dan Double Columnar, vol. 4, no. 5, pp. 31–40, 2015.
[2] T. Zebua, “Encoding the Record Database of Computer Based Test Exam Based on Spritz Algorithm,” Lontar Komput. J. Ilm. Teknol. Inf., vol. 9, no. 1, p. 52, 2018, doi: doi: 10.24843/lkjiti.2018.v09.i01.p06.
[3] B. S. I. M. Arrijal and R. Efendi, “Penerapan Algoritma Kriptografi Kunci Simetris dengan Modifikasi Vigenere Cipher dalam Aplikasi Kriptografi Teks,” J. Pseudocode, vol. 3, no. 1, pp. 69–82, 2016, doi: 10.33369/pseudocode.3.1.69-82.
[4] M. Diana and T. Zebua, “Optimalisasi Beaufort Cipher Menggunakan Pembangkit Kunci RC4 Dalam Penyandian SMS,” J. Sains Komput. Inform., vol. 2, no. 1, pp. 12–22, 2018.
[5] M. B. Sanjaya and P. A. Telnoni, “Implementasi Random Number Blum-Blum-Shub dan Chaotic Function untuk Modifikasi Key Generating pada Kriptografi AES,” J. Elektro Telekomun. Ter. Desember, vol. 1, no. 3, pp. 154–165, 2015.
[6] B. Aïssa, “Implementation of Blum Blum Shub Generator for Message Encryption,” Int. Conf. Control. Enginering Inf. Tecknology (CEIT’ 14), vol. 1, no. 4, p. 2014.
[7] E. Setyaningsih, Kriptografi & Implementasinya Menggunakan Matlab. Yogyakarta: Andi, 2015.
[8] R. A. R. U. Marsal and F. Arnia, “Enkripsi dan Dekripsi Citra Menggunakan Modifikasi Algoritma Vigenere Cipher,” KITEKTRO J. Online Tek. Elektro, vol. 3, no. 3, pp. 6–10, 2018.
[9] H. Mukhtar, Kriptografi untuk Keamanan Data, 1st ed. Yogyakarta: CV. Budi Utama, 2018.
[10] Wikipedia, “Beaufort Cipher,” https://en.wikipedia.org/wiki/ Beaufort_cipher, 2020. [Accessed 17 February 2021].
[11] E. Apriliana, “Kombinasi Algoritma One Time Pad dengan Pembangkit Kunci Blum Blum Shub,” J. Inform. Bandung, vol. 1, no. 70, pp. 11–20, 2016.
[12] A. Hoerudin, “Game Puzzle Makanan Tradisional Dengan Menggunakan Algoritma Blum Blum Shub sebagai Pengacakan Gambar,” J. Fak. Ilmu Komput. Jakarta, vol. 1, no. 6, pp. 1–8, 2010.
[13] T. S. Waruwu and K. Telaumbanua, “Kombinasi Algoritma OTP Cipher dan Algoritma BBS dalam Pengamanan File,” JSM STIMIK Mikroskil, vol. 17, no. 1, pp. 119–126, 2016.
[14] H. Y. A. Sinaga and L. Sitorus, “Pengamanan File Citra Digital Dengan Menggunakan Metode Least Significant Bit Dan End Of File,” J. Tek. Inform. Unika St. Thomas, vol. 2, no. 2, pp. 33–41, 2017.
[15] Aldo and L. Hakim, “Implementasi Steganografi Pada Citra Digital dan Kriptografi Algoritma Hill Cipher Untuk Pengamanan Informasi Berupa Text,” J. Ilm. Teknol. Inf. Terap., vol. V, no. 1, pp. 6–17, 2018.