RandWher: Algoritma Block Chiper dengan Prinsip Random di dalam Jaringan Feistel

Rakhmatullah Yoga Sutrisna
Teknik Informatika
Institut Teknologi Bandung
Bandung, Indonesia
13512053@std.stei.itb.ac.id

Akhmad Fakhoni Listiyan Dede
Teknik Informatika
Institut Teknologi Bandung
Bandung, Indonesia
13513601@std.stei.itb.ac.id

***Abstract*—Makalah ini membahas mengenai rancangan *RandWher block chiper*. Blok chiper ini terinspirasi dari beberapa algoritma blok chiper yang sudah ada, dengan pengembangan enkripsi dan pengacakan pada jaringan Feistel. Enkripsi pada jaringan Feistel dilakukan menggunakan vigenere chiper. Hasil enkripsi diacak sebanyak dua kali sekaligus. Pengacakan yang pertama adalah pengacakan total secara random pada bit yang hasil enkripsi. Setelah diacak total, bit akan di shifting ke kiri. Jumlah shift juga ditentukan secara random. Dengan adanya dua jenis pengacakan ini, maka keamanan algoritma menjadi semakin tinggi.**

Kata kunci—kriptografi; block cipher; bit suffling; bit shifting; random

I. PENDAHULUAN

Penggunaan komputer yang terhubung dalam jaringan internet harus memperhatikan aspek keamanan. Aspek keamanan tersebut meliputi keamanan pengiriman data serta keamanan penyimpanan data dari penyalahgunaan oleh pihak yang tidak berwenang atas data tersebut. Pada aspek inilah kriptografi mengambil peran yang sangat penting. Data yang disimpan atau dikirim harus dalam keadaan terenkripsi sehingga dapat mencegah pihak-pihak yang tidak berwenang untuk mengetahui atau memanipulasi data tersebut.

Kriptografi adalah sebuah ilmu yang mempelajari penyembunyian makna sebuah pesan dengan menggunakan sandi. Kriptografi telah dipelajari semenjak ribuan tahun yang lalu. Hingga saat ini masih banyak sekali riset dan penelitian yang mengembangkan metode kriptografi.

Kriptografi sebelum adanya komputer disebut sebagai kriptografi klasik. Kriptografi ini menggunakan alat seperti *scytale, enigma*, dan sebagainya. Contoh algoritma kriptografi klasik antara lain *vigenere, playfair, caesar*, dan masih banyak lagi. Kriptografi modern muncul setelah datangnya komputer. Berbeda dengan kriptografi klasik, kriptografi modern menggunakan manipulasi bit pada komputer. Manipulasi ini dapat berupa substitusi dan transposisi. Salah satu algoritma dalam kriptografi modern yang hingga saat ini masih menjadi topik penelitian adalah *block cipher*. Algoritma ini melakukan enkripsi pada setiap n blok bit data dimana n adalah jumlah bit kelipatan delapan.

Makalah ini akan membahas algoritma *block cipher* yang kami rancang dengan memanfaatkan beberapa algoritma algoritma enkripsi yang sudah ada, serta mengkombinasikannya dengan fungsi pengacakan bit pada tiap blok. Algoritma ini meningkatkan keamanan dengan kunci yang panjangnya 128 bit.

II. DASAR TEORI

A. *Vigenere Chiper*

Vigenere Chiper digagas oleh kriptologi Prancis bernama Blaise de Vigènere pada abad 16. Prinsip algoritma ini adalah melakukan enkripsi menggunakan bujur sangkar vigenere. [1]

Proses enkripsi E dengan kunci K dan pesan M dilakukan dengan algoritma berikut.

$$Ci = Ek(Mi) = (Mi + Ki) \bmod 26$$

Proses dekripsi D dengan kunci K dilakukan dengan cara membalik proses enkripsi, yaitu

$$Mi = Dk(Ci) = (Ci – Ki) \bmod 26$$

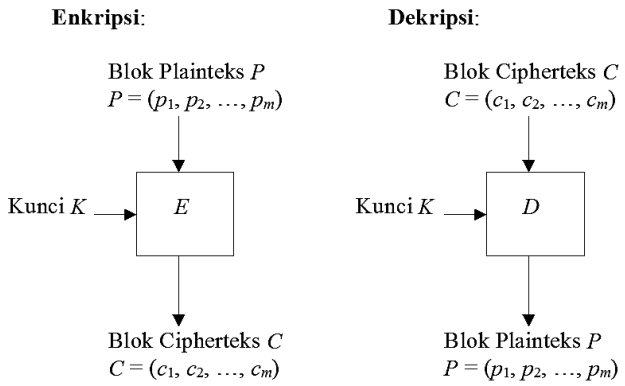
Untuk *vigenere chiper* dengan 255 karakter ASCII, maka modulo 26 diganti menjadi 255, sehingga algoritma vigenere menjadi seperti berikut.

$$Ci = Ek(Mi) = (Mi + Ki) \bmod 256$$

$$Mi = Dk(Ci) = (Ci – Ki) \bmod 256$$

B. *Block Cipher*

Block Cipher adalah algoritma kriptografi modern yang melakukan enkripsi dengan cara membagi plainteks menjadi beberapa blok bit dengan panjang bit yang sama (pada umumnya 64 bit atau kelipatannya). Pada algoritma ini, kunci enkripsi dan dekripsi yang digunakan panjangnya sama dengan panjang blok. Untuk setiap blok plainteks dienkripsi menggunakan kunci yang sama. Dalam proses enkripsi dan dekripsinya, terdapat beberapa modus operasi yang diterapkan oleh algoritma block cipher. Modus-modus tersebut antara lain *Electronic Code Book* (ECB), *Cipher Block Chaining* (CBC), *Cipher Feedback* (CFB), serta *Output Feedback* (OFB). Gambar 1 berikut menjelaskan skema enkripsi dan dekripsi pada *block cipher*.



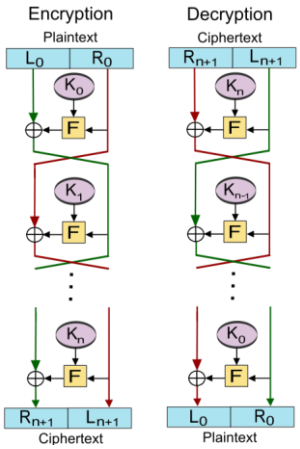
Gambar 1. Skema enkripsi dan dekripsi *block cipher*

C. *Prinsip Diffusion dan Confusion*

Confusion adalah prinsip yang membuat hubungan statistik antara *cipherteks*, *plainteks* dan kunci menjadi sangat rumit. Cara yang dapat dilakukan untuk memenuhi prinsip ini adalah dengan melakukan substitusi yang rumit. Prinsip *Diffusion* adalah prinsip yang menyebarkan pengaruh perubahan sebuah *plainteks* ke sebanyak mungkin *cipherteks*. Prinsip ini digunakan pada modus CBC dan CFB. [2]

D. *Jaringan Feistel*

Jaringan Feistel membuat proses enkripsi pesan menjadi bersifat *reversible*. Sifat *reversible* tersebut membuat perancang algoritma *block cipher* tidak perlu mendesain fungsi balikan untuk melakukan dekripsi. Selain itu sifat *reversible* pada jaringan Feistel juga tidak bergantung pada tingkat kerumitan fungsi enkripsi yang dibuat, sehingga perancang algoritma dapat membuat fungsi yang sedemikian rumit. Untuk proses dekripsi, hanya diperlukan pembalikan potongan *plainteks* beserta urutan kunci internalnya. Skema jaringan Feistel dapat dilihat pada Gambar 2 berikut.



Gambar 2. Jaringan Feistel

E. *Cipher Berulang*

Proses enkripsi dilakukan berulang kali agar mendapatkan *cipherteks* yang rumit. Dalam setiap iterasinya digunakan kunci internal yang unik untuk mengenkripsi hasil dari iterasi sebelumnya.

III. RANCANGAN BLOCK CIPHER

Algoritma block chiper modern ada yang menggunakan S-Box dan shifting. Namun sayangnya, S-Box yang dipakai kebanyak statis, dan tidak berganti-ganti. Jumlah shifting yang dijalankan juga statis. Contohnya pada algoritma GOST, S-Box yang digunakan adalah S-Box statis. Shifting yang dilakukan juga selalu berjumlah 11. Hal ini tentu saja akan mengurangi diferensisasi. Oleh karena itu, rancangan algoritma block chiper ini selalu menggunakan random berdasarkan kunci eksternal pada saat menjalankan Feistel. Random selalu digunakan pada setiap jenis fungsi di Feistel. Oleh karena itu algoritma ini dinamakan RandWher, *Random Everywhere*.

Algoritma block chiper yang dikembangkan menggunakan panjang kunci 128 bit (16 byte), panjang blok 128 bit, dan 16 kali iterasi Feistel. Algoritma ini memiliki 2 bagian utama, yaitu pengacakan kunci dan jaringan Feistel. Pengacakan kunci bertujuan untuk menghilangkan hubungan antara kunci dengan chiper teks. Jaringan Feistel dilakukan dengan kunci internal yang berbeda untuk setiap iterasi. Kunci internal digunakan sebagai kunci enkripsi vigenere. Hasil enkripsi vigenere akan diacak total secara random menggunakan fitur pseudo random java, dengan seed dari kunci internal pada setiap iterasi. Setelah pengacakan, ditambahkan pula *shift bit* ke kiri. Jumlah shift pada setiap iterasi Feistel berbeda tergantung pada hasil random, dengan seed yang berasal dari kunci eksternal.

A. *Pengacakan Kunci*

Pengacakan kunci dilakukan sebanyak 7 kali iterasi. Setiap iterasi, kunci akan dibagi menjadi dua bagian, dan dilakukan pertukaran posisi. Anggap kunci yang digunakan adalah ABCDEFGHIJKLMNOP (128 bit), maka pengacakan dilakukan seperti berikut:

1. ABCDEFGHIJKLMNOP (internal key 1)
2. IJKLMNOPABCDEFGH (internal key 2)
3. MNOPIJKLEFGHABCD (internal key 3)
4. OPMNKLIJGHEFCDAB (internal key 4)
5. PONMLKJIHGFEDCBA (internal key 5)
6. Dan seterusnya hingga terdapat 8 internal key

B. *Pengacakan pada Jaringan Feistel*

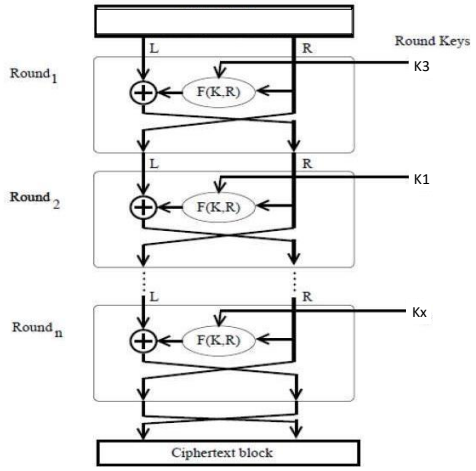
Jaringan Feistel dijalankan sebanyak 16 kali dengan 8 buah kunci internal. Fungsi yang dijalankan pada jaringan Feistel ada 3 jenis:

1. Enkripsi menggunakan vigenere

   Dari hasil pengacakan kunci eksternal, maka didapatkan 8 buah kunci internal. Pada setiap iterasi Feistel, dipilih 1 kunci internal secara random menggunakan seed dari kunci eksternal. Dengan cara ini, kunci internal yang digunakan menjadi tidak memiliki pola tertentu sehingga akan sulit untuk ditebak. Kunci internal digunakan untuk melakukan enkripsi vigenere.

   Misalkan urutan bilangan random yang didapatkan adalah 3,1,2,6,7,5,4, maka pada iterasi pertama, digunakan kunci internal nomor 3, iterasi kedua digunakan kunci internal nomor 1, iterasi ketiga akan

digunakan kunci internal nomor 2, dan seterusnya. Dengan cara seperti ini, 1 buah kunci internal digunakan pada 2 iterasi Feistel yang berbeda.



Gambar 3 Pemanggilan kunci internal yang tidak berurutan, dengan Kx adalah urutan hasil dari random

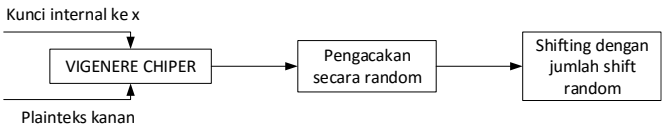
2. Pengacakan semua bit secara random

   Hasil enkripsi dari vigenere diacak total secara random. Pengacakan dilakukan pada setiap iterasi dari Feistel. Seed pada random ini menggunakan kunci eksternal. Dengan adanya random ini, urutan dari bit akan benar-benar teracak

3. *Shifting* bit dengan jumlah shift dinamik

   Setelah pengacakan, hal selanjutnya yang dilakukan adalah melakukan shifting bit. Shifting dilakukan secara sirkular ke kiri. Jumlah shift yang dilakukan bergantung dari nilai random dengan seed kunci eksternal. Banyaknya shift antara 7 – 14 kali.

Jika digambarkan, secara garis besar fungsi F pada jaringan Feistel sebagai berikut.



Gambar 4 Fungsi F pada Feistel

Pada saat melakukan deskripsi, cara yang digunakan membalik jaringan Feistel. Ketika berurusan dengan bilangan random, pengambilan nilai random dilakukan secara terbalik. Misalkan ketika urutan enkripsi menggunakan bilangan random dengan urutan 3,1,2,6,7,5,4, maka ketika melakukan dekripsi, bilangan random harus dibalik menjadi 4,5,7,6,2,1,3. Dengan begini, dekripsi berjalan dengan baik.

IV. Eksperimen dan Pembahasan Hasil

Algoritma RandWher *block cipher* yang telah kami rancang tersebut kemudian kami implementasikan dalam sebuah program kecil dengan bahasa pemrograman Java. Hasil implementasi tersebut selanjutnya dilakukan pengujian. Kasus uji yang dapat dilihat pada Tabel 1 adalah potongan teks sepanjang 64-byte dengan pola berulang setiap 32-byte untuk membuktikan bahwa tidak ada pola berulang pada cipherteks, khususnya pada modus CBC dan CFB 8-bit.

Tabel 1. Kasus uji 1

Plainteks: ABCDEFGHIJKLMNOPQRSTUVWXYZ123456ABCDEFGHIJKLMNOP QRSTUVWXYZ123456 Kunci: KOTABANDUNGJUARA

Potongan teks kasus uji 1 tersebut dienkripsi dengan menggunakan modus ECB, CBC dan CFB 8-bit. Gambar 5, Gambar 6, dan Gambar 7 menunjukkan hasil enkripsi algoritma RandWher.

```
============================
          ECB Mode
============================
Hasil enkripsi:
{□™□,*í?¨.)=
9ÿ□~=â? >fóŒÿy□N×□ö{□™□,*í?¨.)=
9ÿ□~=â? >fóŒÿy□N×□ö

Plainteks hasil dekripsi:
ABCDEFGHIJKLMNOPQRSTUVWXYZ123456ABCDEFGHIJKLMNOPQRSTUVWXYZ123456
```

Gambar 5. Hasil enkripsi kasus uji 1 dengan modus ECB

```
============================
          CBC Mode
============================
Hasil enkripsi:
qþUJĔuö2ST>¬□†'„ᵋG*
□w`M¶´ln¯;L-íš"ê[W'□?È?Ÿ?.¶□‡þÿžc£¬Š¡v¸Ç¹§È¨

Plainteks hasil dekripsi:
ABCDEFGHIJKLMNOPQRSTUVWXYZ123456ABCDEFGHIJKLMNOPQRSTUVWXYZ123456
```

Gambar 6. Hasil enkripsi kasus uji 1 dengan modus CBC

```
============================
          CFB Mode
============================
Hasil enkripsi:
'..."¤□˜3vèJÅ°!ÃglsKƒ`¾Îì^ÐÅ€En¶üÏÞM>gÙèŸŒ□□ðz†□2I?Æ,´¬ å€DÏ□ù:Ò G

Plainteks hasil dekripsi:
ABCDEFGHIJKLMNOPQRSTUVWXYZ123456ABCDEFGHIJKLMNOPQRSTUVWXYZ123456
```

Gambar 7. Hasil enkripsi kasus uji 1 dengan modus CFB 8-bit

Berdasarkan hasil pengujian tersebut, dapat dilihat bahwa plainteks berhasil dienkripsi hingga tidak dikenali lagi maknanya. Selain itu cipherteks juga dapat didekripsi kembali hingga menghasilkan teks yang sama seperti plainteks.

Pada modus ECB cipherteks mempunyai pola berulang sebanyak 32-byte. Hal ini disebabkan oleh cara kerja modus ECB yang melakukan enkripsi per blok tanpa melakukan manipulasi berdasarkan hasil enkripsi pada blok sebelumnya, sehingga terdapat perulangan pada cipherteks. Hal tersebut tidak berlaku pada modus CBC dan CFB 8-bit karena kedua modus

tersebut mengkombinasikan kunci enkripsi pada suatu blok dengan cipherteks hasil enkripsi blok sebelumnya.

V. ANALISIS KEAMANAN

Berdasarkan hasil pengujian algoritma RandWher, kami melakukan analisis keamanan dengan cara membandingkan frekuensi kemunculan karakter pada plainteks dengan frekuensi kemunculan karakter cipherteks. Kasus uji yang kami gunakan untuk analisis ini adalah potongan teks yang kami ambil dari artikel sembarang di media daring. Potongan teks tersebut dapat dilihat pada Tabel 2 berikut.

Tabel 2. Kasus uji 2

```
Plainteks:
Crop Circle Theories

Crop circles are areas of cereal or similar crops
that have been systematically flattened to form
various geometric patterns. The phenomenon itself
only entered the public imagination in its current
form after the notable appearances in England in
the late 1970s. Various scientific and pseudo-
scientific explanations were put forward to
explain the phenomenon, which soon spread around
the world.

In 1991, more than a decade after the phenomena
began, two men, Doug Bower and Dave Chorley,
revealed that they had been making crop circles
in England since 1978 using planks, rope, hats
and wire as their only tools. Many other people
around the world are also openly making crop
circles. Although the commonly accepted view
today is that crop circles are a man-made
phenomenon, paranormal explanations, often
including UFOs, are still popular.

Kunci:
KOTABANDUNGJUARA
```

Hasil enkripsi teks tersebut kemudian dihitung masing-masing frekuensi kemunculan karakternya pada plainteks dan cipherteksnya. Analisis frekuensi pada modus enkripsi ECB, CBC, dan CFB 8-bit dapat dilihat pada Gambar 8, Gambar 9, serta Gambar 10.

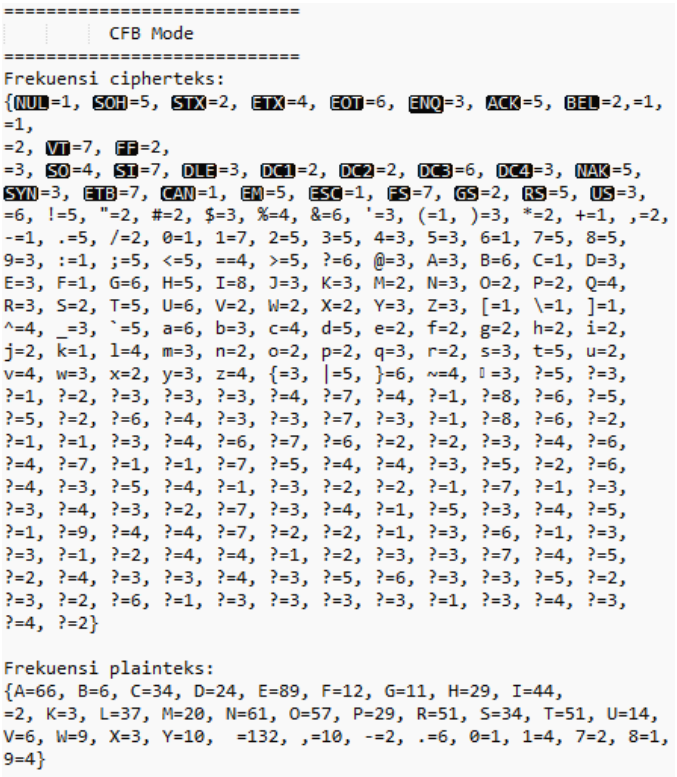
Dari perhitungan frekuensi kemunculan chiper, terlihat bahwa frekuensi kemunculan pada metode CFB lebih berimbang daripada metode ECB dan CBC. Pada modus ECB, masih ada kemiripan jumlah frekuensi. Hal tersebut disebabkan karena pada modus ECB, blok plainteks dioperasikan secara independen, sehingga memungkinkan terjadinya perulangan pola blok cipherteks yang mirip dengan pola perulangan plainteks. Sedangkan pada modus CBC dan CFB 8-bit hasil enkripsi sudah cukup aman karena hubungan statistik antara plainteks, cipherteks dan kunci tidak dapat diperkirakan.

```
============================
        ECB Mode
============================
Frekuensi cipherteks:
{NUL=3, SOH=2, STX=1, ETX=5, EOT=1, ENQ=7, ACK=2, BEL=6,=2,
=5,
=3, VT=3, FF=1,
=5, SO=2, SI=1, DLE=4, DC1=3, DC2=4, DC3=3, DC4=5, NAK=3,
SYN=4, ETB=5, CAN=2, EM=2, SUB=4, ESC=2, FS=3, GS=7, RS=5,
US=2,  =5, !=2, "=4, $=3, %=3, &=5, '=3, (=4, )=5, *=4, +=8, -=1,
.=3, /=1, 0=2, 1=6, 2=9, 3=2, 4=5, 5=1, 6=4, 7=3, 8=3, 9=2, :=2,
;=6, <=2, >=5, ?=7, @=4, A=5, B=3, C=4, D=3, E=1, F=1, G=3, H=2,
I=6, J=5, K=2, L=2, M=6, N=1, O=2, P=2, Q=2, R=3, S=6, U=3, V=5,
W=3, X=4, Y=6, Z=6, [=4, \=5, ]=6, ^=2, _=3, `=4, a=2, b=5, c=8,
d=7, e=4, f=1, g=3, h=1, i=1, j=3, k=7, l=4, m=2, n=4, o=2, p=1,
r=3, s=3, t=2, u=4, v=3, w=4, x=2, y=8, z=6, |=4, }=2, ~=3, ▯=4,
?=4, ?=5, ?=1, ?=1, ?=3, ?=8, ?=4, ?=2, ?=2, ?=1, ?=5, ?=4, ?=3,
?=5, ?=8, ?=2, ?=3, ?=4, ?=2, ?=3, ?=4, ?=4, ?=4, ?=4, ?=3, ?=7,
?=4, ?=6, ?=2, ?=4, ?=3, ?=2, ?=5, ?=2, ?=6, ?=2, ?=5, ?=3, ?=3,
?=6, ?=2, ?=3, ?=1, ?=5, ?=1, ?=2, ?=4, ?=2, ?=2, ?=5, ?=4, ?=4,
?=1, ?=4, ?=2, ?=3, ?=6, ?=4, ?=3, ?=4, ?=3, ?=5, ?=1, ?=2, ?=5,
?=2, ?=3, ?=1, ?=2, ?=5, ?=3, ?=6, ?=3, ?=2, ?=1, ?=9, ?=2, ?=4,
?=2, ?=2, ?=3, ?=3, ?=4, ?=4, ?=7, ?=4, ?=3, ?=2, ?=1, ?=5, ?=3,
?=4, ?=3, ?=4, ?=3, ?=3, ?=3, ?=3, ?=2, ?=7, ?=3, ?=5, ?=6, ?=3,
?=4, ?=2, ?=1, ?=3, ?=2, ?=3, ?=2, ?=4, ?=3, ?=5, ?=4, ?=7, ?=4,
?=2, ?=5, ?=4, ?=1, ?=7, ?=4, ?=1}

Frekuensi plainteks:
{A=66, B=6, C=34, D=24, E=89, F=12, G=11, H=29, I=44,
=2, K=3, L=37, M=20, N=61, O=57, P=29, R=51, S=34, T=51, U=14,
V=6, W=9, X=3, Y=10,  =132, ,=10, -=2, .=6, 0=1, 1=4, 7=2, 8=1,
9=4}
```

Gambar 8. Hasil analisis frekuensi untuk kasus uji 2 modus ECB

```
============================
        CBC Mode
============================
Frekuensi cipherteks:
{NUL=7, SOH=5, STX=1, ETX=6, EOT=2, ENQ=2, ACK=7, BEL=3,=4,
=3,
=3, VT=3, FF=4,
=5, SO=3, SI=3, DLE=2, DC1=3, DC2=2, DC3=4, DC4=5, NAK=2,
SYN=4, ETB=7, CAN=6, EM=5, SUB=8, ESC=4, FS=2, GS=5, US=6,
=2, "=4, #=1, $=2, %=4, &=2, '=4, (=1, )=3, *=2, +=2, ,=3, -=6,
.=5, /=2, 0=5, 1=2, 2=3, 3=5, 4=2, 5=4, 6=3, 7=3, 8=4, 9=5,
:=5, ;=2, <=3, ==2, >=2, ?=8, @=2, A=2, B=3, C=3, D=5, E=1,
F=4, G=1, H=5, I=7, J=4, K=3, L=4, M=3, N=3, O=3, P=4, Q=2,
R=3, S=3, T=3, U=4, V=2, W=4, X=2, Z=4, [=4, \=6, ]=3, ^=8,
_=5, `=4, a=1, b=2, c=2, d=6, e=2, f=4, g=3, h=4, i=4, j=4,
k=4, l=3, m=2, n=3, o=2, p=2, q=3, r=5, s=3, t=4, u=3, v=2,
w=3, x=8, y=2, z=2, {=3, |=4, }=4, ~=2, ▯=3, ?=3, ?=3, ?=8,
?=5, ?=2, ?=2, ?=4, ?=5, ?=6, ?=9, ?=3, ?=5, ?=1, ?=5, ?=1,
?=7, ?=4, ?=3, ?=2, ?=2, ?=2, ?=1, ?=4, ?=2, ?=7, ?=5, ?=6,
?=1, ?=8, ?=5, ?=3, ?=4, ?=4, ?=3, ?=3, ?=4, ?=2, ?=6, ?=1,
?=4, ?=4, ?=2, ?=2, ?=1, ?=3, ?=3, ?=5, ?=3, ?=1, ?=2, ?=4,
?=1, ?=3, ?=3, ?=3, ?=1, ?=1, ?=1, ?=2, ?=2, ?=3, ?=5, ?=1,
?=5, ?=8, ?=3, ?=3, ?=5, ?=2, ?=3, ?=4, ?=2, ?=7, ?=4, ?=1,
?=4, ?=3, ?=3, ?=1, ?=2, ?=7, ?=1, ?=4, ?=4, ?=3, ?=3, ?=4,
?=3, ?=5, ?=3, ?=2, ?=8, ?=4, ?=7, ?=6, ?=1, ?=3, ?=3, ?=2,
?=4, ?=4, ?=1, ?=2, ?=1, ?=1, ?=9, ?=2, ?=2, ?=2, ?=1, ?=3,
?=6, ?=6, ?=3, ?=5, ?=4, ?=4, ?=2, ?=3, ?=2, ?=3, ?=3, ?=1,
?=4, ?=2}

Frekuensi plainteks:
{A=66, B=6, C=34, D=24, E=89, F=12, G=11, H=29, I=44,
=2, K=3, L=37, M=20, N=61, O=57, P=29, R=51, S=34, T=51, U=14,
V=6, W=9, X=3, Y=10,  =132, ,=10, -=2, .=6, 0=1, 1=4, 7=2, 8=1,
9=4}
```

Gambar 9. Hasil analisis frekuensi untuk kasus uji 2 modus CBC



```
============================
        CFB Mode
============================
Frekuensi cipherteks:
{NUL=1, SOH=5, STX=2, ETX=4, EOT=6, ENQ=3, ACK=5, BEL=2,=1,
=1,
=2, VT=7, FF=2,
=3, SO=4, SI=7, DLE=3, DC1=2, DC2=2, DC3=6, DC4=3, NAK=5,
SYN=3, ETB=7, CAN=1, EM=5, ESC=1, FS=7, GS=2, RS=5, US=3,
=6, !=5, "=2, #=2, $=3, %=4, &=6, '=3, (=1, )=3, *=2, +=1, ,=2,
-=1, .=5, /=2, 0=1, 1=7, 2=5, 3=5, 4=3, 5=3, 6=1, 7=5, 8=5,
9=3, :=1, ;=5, <=5, ==4, >=5, ?=6, @=3, A=3, B=6, C=1, D=3,
E=3, F=1, G=6, H=5, I=8, J=3, K=3, M=2, N=3, O=2, P=2, Q=4,
R=3, S=2, T=5, U=6, V=2, W=2, X=2, Y=3, Z=3, [=1, \=1, ]=1,
^=4, _=3, `=5, a=6, b=3, c=4, d=5, e=2, f=2, g=2, h=2, i=2,
j=2, k=1, l=4, m=3, n=2, o=2, p=2, q=3, r=2, s=3, t=5, u=2,
v=4, w=3, x=2, y=3, z=4, {=3, |=5, }=6, ~=4, =3, ?=5, ?=3,
?=1, ?=2, ?=3, ?=3, ?=3, ?=4, ?=7, ?=4, ?=1, ?=8, ?=6, ?=5,
?=5, ?=2, ?=6, ?=4, ?=3, ?=3, ?=7, ?=3, ?=1, ?=8, ?=6, ?=2,
?=1, ?=1, ?=3, ?=4, ?=6, ?=7, ?=6, ?=2, ?=2, ?=3, ?=4, ?=6,
?=4, ?=7, ?=1, ?=1, ?=7, ?=5, ?=4, ?=4, ?=3, ?=5, ?=2, ?=6,
?=4, ?=3, ?=5, ?=4, ?=1, ?=3, ?=2, ?=2, ?=1, ?=7, ?=1, ?=3,
?=3, ?=4, ?=3, ?=2, ?=7, ?=3, ?=4, ?=1, ?=5, ?=3, ?=4, ?=5,
?=1, ?=9, ?=4, ?=4, ?=7, ?=2, ?=2, ?=1, ?=3, ?=6, ?=1, ?=3,
?=3, ?=1, ?=2, ?=4, ?=4, ?=1, ?=2, ?=3, ?=3, ?=7, ?=4, ?=5,
?=2, ?=4, ?=3, ?=3, ?=4, ?=3, ?=5, ?=6, ?=3, ?=3, ?=5, ?=2,
?=3, ?=2, ?=6, ?=1, ?=3, ?=3, ?=3, ?=3, ?=1, ?=3, ?=4, ?=3,
?=4, ?=2}

Frekuensi plainteks:
{A=66, B=6, C=34, D=24, E=89, F=12, G=11, H=29, I=44,
=2, K=3, L=37, M=20, N=61, O=57, P=29, R=51, S=34, T=51, U=14,
V=6, W=9, X=3, Y=10,  =132, ,=10, -=2, .=6, 0=1, 1=4, 7=2, 8=1,
9=4}
```

Gambar 10. Hasil analisis frekuensi untuk kasus uji 2 modus CFB 8-bit

VI. KESIMPULAN DAN SARAN

Berdasarkan hasil analisis, algoritma RandWher dapat melakukan enkripsi dengan baik. Keterhubungan antara kunci dengan teks chiper sulit untuk dilakukan. Pengembangan lebih lanjut dapat dilakukan dengan cara melakukan random terhadap kunci eksternal, dan juga penambahan fungsi random pada fungsi Feistel.

DAFTAR REFERENSI

[1] R. Munir, Slide Kuliah Kriptografi, Bandung: Teknik Informatika ITB, 2015.

[2] C. Shannon, "Communication theory of secrecy systems," *Bell Systems Technical Journal 28*, p. 656 – 715, 1949.