**T-3**

# PENGGUNAAN POLINOMIAL UNTUK STREAM KEY GENERATOR PADA ALGORITMA STREAM CIPHERS BERBASIS FEEDBACK SHIFT REGISTER

**Arga Dhahana Pramudianto[1], Rino[2]**

[1,2]Sekolah Tinggi Sandi Negara
arga.daywalker@gmail.com, rinosebastian05@gmail.com

**Abstrak**

Karya tulis yang berjudul **"Penggunaan Polinomial untuk Stream Key Generator pada Algoritma Stream Ciphers Berbasis Feedback Shift Register"** ini kami tulis dalam rangka memperkenalkan salah satu algoritma sandi yang memanfaatkan kontribusi ilmu Matematika didalamnya. Perkembangan ilmu pengetahuan dan teknologi informasi dan komunikasi yang sedemikian pesat dewasa ini, telah dirasakan manfaatnya dalam kehidupan dunia yang modern, namun sangat jarang kita berfikir dampak negatif dan faktor pengamanannya dalam rangka menjamin kerahasiaan suatu informasi agar tidak jatuh ke pihak yang tidak berkepentingan. Hal ini telah mendorong perkembangan ilmu pengetahuan dan teknologi persandian yang semakin maju dan kompleks seiring dengan perkembangan ilmu pengetahuan dan teknologi informasi dan komunikasi yang ada sekarang ini.

Ilmu persandian / kriptografi adalah ilmu yang erat kaitannya dengan ilmu Matematika. Kontribusi ilmu Matematika dalam bidang persandian cukup mendukung dalam rangka terciptanya suatu algoritma sandi modern. Salah satunya adalah penggunaan polinomial untuk pembangkit kunci (key generator) pada algoritma stream chipers.

Polinomial digunakan sebagai penentu koordinat operasi logika dalam pembangkit kunci (key generator). Output dari pembangkit kunci (key generator) adalah berupa rangkaian bit untuk selanjutnya dilakukan operasi penyandian (enkripsi) pada kode ASCII setiap karakter teks terang (plain text) secara bit per bit. Hasil penyandian atau yang disebut cipher text adalah berupa rangkaian bit sandi yang akan membentuk kode ASCII baru untuk selanjutnya diterjemahkan menjadi karakter baru.

**Kata kunci**: polinomial, feedback shift register, stream cipher

## PENDAHULUAN

Berkat perkembangan teknologi yang begitu pesat memungkinkan manusia dapat berkomunikasi dan saling bertukar informasi/data secara jarak jauh. Antar kota antar wilayah antar negara bahkan antar benua bukan merupakan suatu kendala lagi dalam melakukan komunikasi dan pertukaran data. Seiring dengan itu tuntutan akan sekuritas (keamanan) terhadap kerahasiaan informasi yang saling dipertukarkan tersebut semakin meningkat. Begitu banyak pengguna seperti departemen pertahanan, suatu perusahaan atau bahkan individu-individu tidak ingin informasi yang disampaikannya diketahui oleh

orang lain atau kompetitornya atau negara lain. Oleh karena itu dikembangkanlah cabang ilmu yang mempelajari tentang cara-cara pengamanan data atau dikenal dengan istilah Kriptografi.

## PEMBAHASAN

Dalam kriptografi terdapat dua konsep utama yakni enkripsi dan dekripsi. Enkripsi adalah proses dimana informasi/data yang hendak dikirim diubah menjadi bentuk yang hampir tidak dikenali sebagai informasi awalnya dengan menggunakan algoritma tertentu. Dekripsi adalah kebalikan dari enkripsi yaitu mengubah kembali bentuk tersamar tersebut menjadi informasi awal.

Algoritma kriptografi berdasarkan jenis kunci yang digunakan dapat dibedakan menjadi dua jenis yaitu :

- **Algoritma Simetris**

  Algoritma simetris *(symmetric algorithm)* adalah suatu algoritma dimana kunci enkripsi yang digunakan sama dengan kunci dekripsi sehingga algoritma ini disebut juga sebagai *single-key algorithm.*

- **Algoritma Asimetris**

  Algoritma asimetris *(asymmetric algorithm)* adalah suatu algoritma dimana kunci enkripsi yang digunakan tidak sama dengan kunci dekripsi. Pada algoritma ini menggunakan dua kunci yakni kunci publik *(public key)* sebagai kunci enkripsi dan kunci privat *(private key)*sebagai kunci dekripsi. Kunci publik disebarkan secara umum sedangkan kunci privat disimpan secara rahasia oleh pengguna (user) yang berkepentingan. Walau kunci publik telah diketahui namun akan sangat sukar mengetahui kunci privat yang digunakan.

Sedangkan berdasarkan besar data yang diolah dalam satu kali proses, maka algoritma kriptografi dapat dibedakan menjadi dua jenis yaitu :

- **Algoritma Block Cipher**

Informasi/data yang hendak dikirim dalam bentuk blok-blok besar (misal 64-bit) dimana blok-blok ini dioperasikan dengan fungsi enkripsi yang sama dan akan menghasilkan informasi rahasia dalam blok-blok yang berukuran sama.

- **Algoritma Stream Cipher**

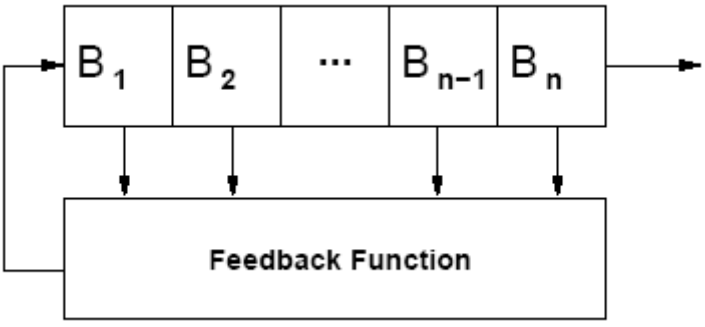
  Informasi/data yang hendak dikirim dioperasikan dalam bentuk blok-blok yang lebih kecil (byte atau bit), biasanya satu karakter persatuan waktu proses, menggunakan tranformasi enkripsi yang berubah setiap waktu.

Peranan kunci sangatlah penting dalam proses enkripsi dan dekripsi (disamping algoritma yang digunakan). Pada algoritma stream cipher yang akan kami perkenalkan ini menggunakan feedback shift register sebagai pembangkit aliran kunci dan polinomial untuk menentukan koordinat operasi logika dalam pembangkitan kunci berikut periode atau panjang output bit kuncinya.

Register geser umpan-balik (*feedback shift register*) atau *FSR* memiliki dua bagian utama, yaitu :

1. Register geser atau *shift register*

   Berupa rangkaian atau barisan bit-bit($B_n, B_{n-1}, \dots, B_2, B_1$) yang panjangnya *n* (disebut juga *shift register n*-bit), Isi dari *shift register* biasanya berupa bilangan biner, yaitu angka 1 dan angka 0.
2. Fungsi umpan balik atau *feedback function* yaitu fungsi yang menerima masukan dari register geser dan mengembalikan nilai fungsi ke register geser.




Gambar 1 : Bagian-bagian FSR

Setiap kali sebuah bit dibutuhkan, semua bit yang ada di dalam register akan bergeser 1 bit ke kanan atau ke kiri tergantung dari bagaimana perancangan *stream key generator*-nya. Posisi bit yang berada di ujung kanan atau ujung kiri register dihitung sebagai fungsi bit-bit lain di dalam register tersebut. Keluaran dari register geser adalah 1 bit (yaitu bit yang tergeser keluar dari kanan atau kiri register). **Periode** shift register adalah panjang barisan keluaran (output) sebelum ia berulang kembali.

Contoh *feedback shift register* adalah *linear feedback shift register* atau *LFSR*. Dimana *feedback function*-nya adalah peng-XOR-an bit-bit tertentu di dalam register. Dengan demikian, LFSR merupakan shift register yang output bitnya dibangkitkan dari proses XOR *seed bit* atau nilai bit awal yang ada pada register dengan panjang tertentu tergantung dari derajat polinomial yang digunakan.

Dalam matematika, polinomial atau suku banyak adalah pernyataan matematika yang melibatkan jumlahan perkalian pangkat dalam satu atau lebih variabel dengan koefisien. Sebuah polinomial dalam satu variabel dengan koefisien konstan memiliki bentuk seperti berikut:

$$a_n x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \ldots + a_2 x^2 + a_1 x + a_0 ; a_n \neq 0, n \in Z^+$$

- $a_n, a_{n-1}, a_{n-2}, \ldots, a_1$ disebut koefisien-koefisien polinomial dari masing-masing variable x yang merupakan konstanta real, dan $a_n \neq 0$.
- $a_0$ disebut suku tetap (konstanta).
- Pangkat tertinggi pada suatu polinomial menunjukkan orde atau derajat dari polinomial tersebut.

**Grafik Polinomial**

Sebuah fungsi polinomial dalam satu variabel real dapat dinyatakan dalam grafik fungsi.

- ***Grafik dari polinomial nol***

  f(x) = 0 adalah sumbu x.

- ***Grafik dari polinomial berderajat nol***

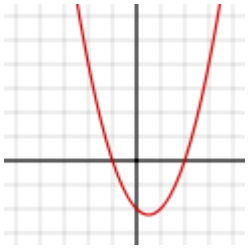
  f(x) = $a_0$, dimana $a_0 \neq 0$, adalah garis horizontal dengan y memotong $a_0$.

- ***Grafik dari polinomial berderajat satu (atau fungsi linear)***

  $f(x) = a_0 + a_1x$ , dengan $a_1 \neq 0$, adalah berupa garis miring dengan y memotong di $a_0$ dengan kemiringan sebesar $a_1$.

- ***Grafik dari polinomial berderajat dua***

  $f(x) = a_0 + a_1x + a_2x^2$, dengan $a_2 \neq 0$ adalah berupa parabola.



Gambar 2 : Grafik Polinomial Berderajat 2

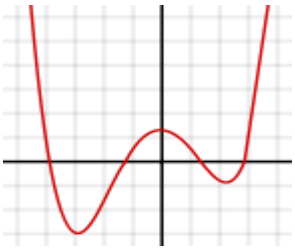
- ***Grafik dari polinomial berderajat tiga***

  $f(x) = a_0 + a_1x + a_2x^2$, $+ a_3x^3$, dengan $a_3 \neq 0$ adalah berupa kurva pangkat 3.

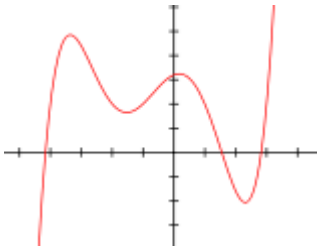
Gambar 3 : Grafik Polinomial Berderajat 3

- ***Grafik dari polinomial berderajat dua atau lebih***

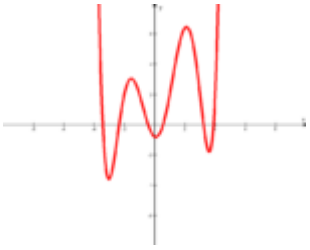
  $f(x) = a_0 + a_1x + a_2x^2 + ... + a_nx^n$ , dengan $a_n \neq 0$ dan $n \geq 2$
  adalah berupa kurva non-linear.



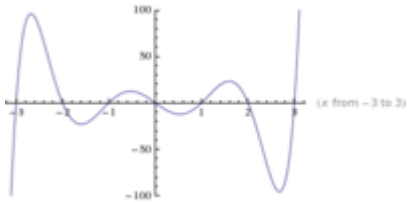
Gambar 4 : Grafik Polinomial Berderajat 4



Gambar 5 : Grafik Polinomial Berderajat 5



Gambar 6 : Grafik Polinomial Berderajat 6




Gambar 7 : Grafik Polinomial Berderajat 7

Agar periode pada *LFSR* maksimum, maka polinomial yang digunakan pada *stream key generator* harus merupakan polinomial primitif, atau polinomial sembarang yang telah diuji keprimitifannya. Setiap derajat pada polinomial, bisa memiliki beberapa polinomial primitif yang berbeda.

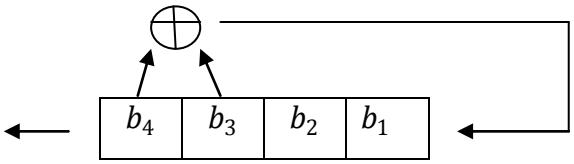
Contohnya :

$x^4 + x^3 + 1$ adalah polinomial berderajat 4, reciprocal dari polinomial tersebut adalah $x^4 + x + 1$ dimana keduanya merupakan polinomial primitif.

Pada umumnya, tidak ada cara yang mudah dalam pembangkitan polinomial primitif untuk derajat sembarang yang diberikan. Cara yang paling mudah adalah dengan memilih polinomial secara acak dan menguji polinomial tersebut primitif atau tidak. Pada contoh polinomial primitif diatas [4, 3, 0] , angka pertama adalah panjang dari *LFSR*-nya, sedangkan angka terakhir akan selalu 0 atau bisa diabaikan.

Pada contoh tersebut juga mengandung artian bahwa *seed bit* kita atau nilai bit semula kita dalam register memiliki panjang 4 bit dan untuk membangkitkan bit baru adalah dengan meng-XOR-kan bit keempat dan ketiga. Periode dari *LFSR*-nya akan maksimum karena polinomial yang digunakan adalah polinomial primitif, output bitnya akan berulang sebelum $2^4 - 1$.

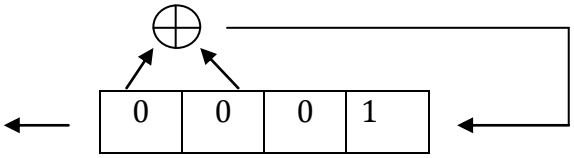
Periode maksimum *LFSR n*-bit memiliki rumus umum $2^n - 1$, bukan $2^n$ karena akan menghasilkan barisan bit 0 yang tidak pernah berakhir, (0000 0000 0000 ....) yang tidak berguna sebagai isi dalam suatu register.




Gambar 8 : LFSR-4 stages

Berikut ini adalah contoh pengujian polinomial untuk menghasilkan periode *LFSR* yang maksimum. *Feedback function* yang digunakan adalah sama-sama berderajat 4, yaitu :

1. $f(x) = x^4 + x^3 + 1$ dengan seed : 0001

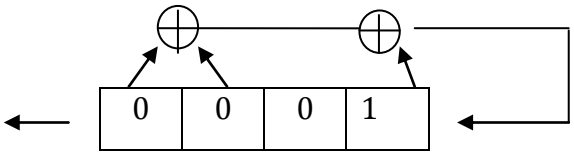



Gambar 9 : Pengujian polinomial-1

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |

menghasilkan bit key stream : 0001 0011 0101 111 dengan periode 15 (sesuai dengan rumus $2^4 - 1$), sehingga dapat disimpulkan bahwa polinomial tersebut adalah polinomial primitif karena menghasilkan periode *LFSR* yang maksimum.

2. $f(x) = x^4 + x^3 + x + 1$ dengan seed : 0001




Gambar 10 : Pengujian polinomial-2

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |

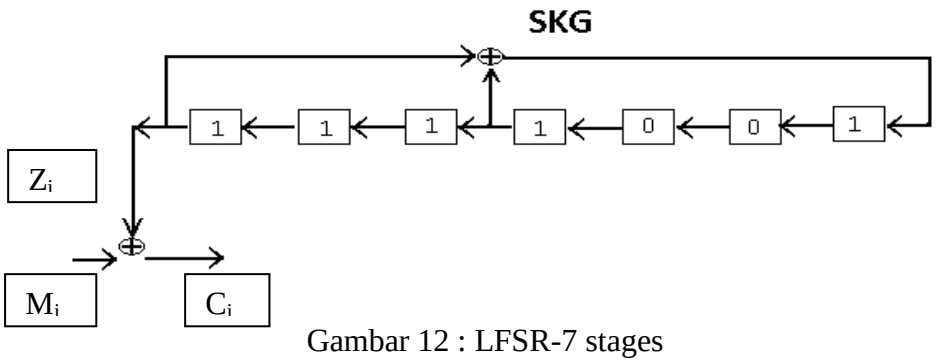
menghasilkan bit key stream 0001 11 dengan periode 6, sehingga dapat disimpulkan bahwa polinomial tersebut tidak primitif karena tidak menghasilkan periode *LFSR* yang maksimum.

Berikut ini adalah tabel beberapa polinomial primitif yang sudah pasti akan menghasilkan periode *LFSR* maksimum :

| Bits ( $n$ ) | Polinomial Primitif | Periode ( $2^n-1$ ) |
|---|---|---|
| 2 | $x^2+x+1$ | 3 |
| 3 | $x^3+x^2+1$ | 7 |
| 4 | $x^4+x^3+1$ | 15 |
| 5 | $x^5+x^4+1$ | 31 |
| 6 | $x^6+x^5+1$ | 63 |
| 7 | $x^7+x^6+1$ | 127 |
| 8 | $x^8+x^6+x^5+x^4+1$ | 255 |
| 9 | $x^9+x^5+1$ | 511 |
| 10 | $x^{10}+x^7+1$ | 1023 |
| 11 | $x^{11}+x^9+1$ | 2047 |
| 12 | $x^{12}+x^{11}+x^{10}+x^4+1$ | 4095 |
| 13 | $x^{13}+x^{12}+x^{11}+x^8+1$ | 8191 |
| 14 | $x^{14}+x^{13}+x^{12}+x^2+1$ | 16383 |
| 15 | $x^{15}+x^{14}+1$ | 32767 |
| 16 | $x^{16}+x^{14}+x^{13}+x^{11}+1$ | 65535 |
| 17 | $x^{17}+x^{14}+1$ | 131071 |
| 18 | $x^{18}+x^{11}+1$ | 262143 |
| 19 | $x^{19}+x^{18}+x^{17}+x^{14}+1$ | 524287 |
| dst | dst | dst |
| | | |

Gambar 11 : Tabel Polinomial Primitif

Berikut ini adalah implementasi sederhana penggunaan output rangkaian bit dari *Stream Key Generator (SKG)* yang memanfaatkan *LFSR* untuk penyandian (*enkripsi*) suatu teks terang (*plain-text*) dengan fungsi XOR. Untuk menghasilkan periode maksimum pada barisan outputnya, maka fungsi polinomial yang digunakan harus primitif. Kita ambil sebuah *LFSR-7 stages* dengan *feedback function : f(x)= $x^7 + x^4 + 1$* dan *seed bit* atau nilai bit semulanya adalah 1111001 yang akan mengisi *stage-stage* $S_0$, $S_1$,….,$S_6$ secara berurutan, diagram pembangkit aliran kunci (*SKG*) tersebut ditunjukan pada gambar dibawah ini :




Gambar 12 : LFSR-7 stages

Dari *LFSR* di atas sebagai *SKG*, di peroleh output rangkaian bit dengan periode maksimum yaitu $2^7 - 1 = 127$ sebagai berikut:

11110 01011 00100 10000 00100 01001

10001 01110 10110 11000 00110 01101

01001 11001 11101 10100 00101 01011

11101 00101 00011 01110 00111 11110

00011 10

Misal terdapat potongan teks terang **STREAM CIPHER** yang akan di enkripsi dengan rangkaian bit diatas sebagai kuncinya, tahap pertama adalah mengkonversi potongan teks terang tersebut ke dalam bentuk biner. Teks **STREAM CIPHER** dikonversi ke bentuk *ASCII*-nya menjadi : **83 84 82 69 65 77 67 73 80 72 69 82**. Jika bit terakhir adalah bit paritas untuk cek kesalahan dengan menghitung $\sum 1 \bmod 2$, maka hasil konversi ke binernya adalah:

10100110 10101001 10100101 10001011

10000010 10011010 10000111 10010011

10100000 10010000 10001011 10100101

Akan menghasilkan bit-bit sandi sebagai berikut :

Tt : 10100110 10101001 10100101 10001011 10000010 10011010

Rk : 11110010 11001001 00000010 00100110 00101110 10110110 ⊕

Ts : 01010100 01100000 10100111 10101101 10101100 00101100

Tt : 10000111 10010011 10100000 10010000 10001011 10100101

Rk : 00001100 11010100 11100111 10110100 00101010 11111010 ⊕

Ts : 10001011 01000111 01000111 00100100 10100001 01011111

Hasil *enkripsi* dengan fungsi XOR adalah : **84 96 167 173 172 44 139 71 71 36 161 79** (dalam nilai desimal), selanjutnya dikonversi menurut tabel *ASCII* untuk memperoleh teks sandi sebagai berikut:

**T ` § - ¬ , PLD(Partial line down) G G $ ¡ O**

Untuk proses *dekripsi*, dilakukan kebalikannya dengan meng-XOR-kan bit teks sandi dengan rangkaian bit kunci yang diperoleh dari pembangkit aliran kunci (*SKG*). Selanjutnya dengan mengabaikan bit paritas pada *least significant bit* (bit ke-8), dilakukan konversi dengan table ASCII untuk mendapatkan teks terang.

Teks sandi **T `§ - ¬ , PLD(Partial line down) G G $ ¡ O** dikonversi ke dalam bentuk *ASCII*-nya menjadi : **84 96 167 173 172 44 139 71 71 36 161 79**

Setelah mendapatkan nilai *ASCII* dari teks sandi tersebut, kemudian dikonversi ke dalam bentuk rangkaian bit seperti di bawah ini:

01010100 01100000 10100111 10101101

10101100 00101100 10001011 01000111

01000111 00100100 10100001 01011111

Maka akan menghasilkan bit-bit terang sebagai berikut :

Ts : 01010100 01100000 10100111 10101101 10101100 00101100

Rk : 11110010 11001001 00000010 00100110 00101110 10110110 ⊕

Tt : 10100110 10101001 10100101 10001011 10000010 10011010

Ts : 10001011 01000111 01000111 00100100 10100001 01011111

Rk : 00001100 11010100 11100111 10110100 00101010 11111010 ⊕

Tt : 10000111 10010011 10100000 10010000 10001011 10100101

Setelah mendapatkan bit-bit terang, dengan mengabaikan bit paritas pada *least significant bit* (bit ke-8), menjadi seperti berikut:

1010011 1010100 1010010 1000101

1000001 1001101 1000011 1001001

1010000 1001000 1000101 1010010

kemudian dikonversi ke dalam bentuk desimal, yaitu: **83 84 82 69 65 77 67 73 80 72 69 82.** selanjutnya dikonversi menurut tabel *ASCII* untuk memperoleh teks terang sebagai berikut :

**S T R E A M C I P H E R**

Pada algoritma diatas, dengan menggunakan sebuah *LFSR*-7 *stages*, maka jumlah variasi kemungkinan kunci input (periode maksimum) adalah $2^7$ - 1 (kecuali kunci input nol semua). Setiap kunci input akan menghasilkan output rangkaian bit kunci yang berbeda, sehingga jumlah total kemunginan output juga sama yaitu $2^7$ - 1. Karena *feedback function* yang digunakan adalah polinomial karakteristik yang primitif, maka dapat dijamin keseluruhan barisan outputnya mempunyai periode yang maksimum yaitu $2^7$ - 1.

## KESIMPULAN

Kriptografi adalah ilmu yang mempelajari tentang cara-cara pengamanan data. Algoritma kriptografi berdasarkan jenis kunci yang digunakan dapat dibedakan menjadi dua jenis yaitu algoritma simetris dan algoritma asimetris, sedangkan menurut besar data yang diolah dalam satu kali proses juga terdapat dua jenis, yaitu stream ciphers dan block ciphers. Pembangkitan aliran kunci pada algoritma stream ciphers banyak memanfaatkan penggunaan *feedback shift register* karena mudah diimplementasikan ke dalam hardware digital. *Feedback shift register* terdiri dari *shift register* (berupa bilangan biner) dan *feedback function* (operasi XOR didalam register). Contoh *feedback shift register* adalah *linear feedback shift register* atau *LFSR*. Agar periode *LFSR* yang dihasilkan maksimum, maka diperlukan adanya polinomial primitif atau polinomial sembarang yang telah diuji keprimitifannya menurut rumus umum $2^n - 1$. Dimana *n* adalah derajat tertinggi dalam suatu polinomial yang digunakan. Selain itu penggunaan polinomial pada *feedback shift register* adalah juga untuk menentukan koordinat operasi logika (XOR) pada *feedback function*-nya. Output dari pembangkit aliran kunci (stream key generator) berbasis *feedback shift register* tersebut adalah berupa rangkaian bit untuk selanjutnya digunakan dalam operasi penyandian (enkripsi) pada setiap karakter teks terang (plain text) secara bit per bit. Dan akan menghasilkan bit sandi yang kemudian diterjemahkan menjadi teks sandi (cipher text) dengan tingkat kerahasiaan tertentu.

## DAFTAR PUSTAKA

Bruce Schneier. 1996. *Applied Cryptography*. New York: John Wiley & Sons, Inc.

LEMSANEG RI. 2007. *Jelajah Kriptologi*. Jakarta: Lembaga Sandi Negara.

Rinaldi Munir. 2004. *Kriptografi*. Bandung: Penerbit INFORMATIKA.

http://id.wikipedia.org/wiki/Polinomial

http://www.google.com