KESATRIA: Jurnal Penerapan Sistem Informasi (Komputer & Manajemen)
Terakreditasi Nomor 204/E/KPT/2022 | Vol. 4, No. 2, April (2023), pp. 409-419

Desain Pembangkit Kunci LFSR dengan Skema A5/1 Menggunakan empat Blok Bit Fungsi XOR

Teguh Septianto Nahading[1], Alz Danny Wowor[2]
[1,2]Teknik Informatika, FTI UKSW, Salatiga, Jawa Tengah, Indonesia
E-mail: [1]672022709@student.uksw.edu, [2]alzdanny.wowor@uksw.edu

Abstract

This research designs the Random Number Generator by using the LFSR with an A5/1 scheme and also four feedback for the Random Number Generator. Every feedback functions use the XOR operation to determine the new bit value in the next literation. Random results of the Run Test, Mono Bit, and Block bit shows that the algorithm is consistently able to generate random output even with various input variant. Comparison examination using previous research algorithms shows that four selected feedback functions generate random numbers. The encryption test shows that plaintext and ciphertext have "very low" and "low" correlation rates where the average correlation for every plaintext test is close to 0. The LFSR key generator with an A5/1 algorithm scheme using four XOR functions can generate very good random output for use in Stream Cipher.

Keywords: *Scheme A5/1, Stream Cipher, Linear Feedback Shift Register.*

Abstrak

Penelitian ini merancang pembangkit bilangan acak pendekatan LFSR dengan skema A5/1 dan empat fungsi umpan balik untuk proses pembangkit bilangan acak. Setiap fungsi umpan balik digunakan operasi XOR untuk menentukan nilai bit yang baru pada iterasi berikutnya. Hasil pengujian keacakan Run test, Mono bit, dan Block bit diperoleh bahwa algoritma secara konsisten mampu menghasilkan luaran yang acak walaupun dengan berbagai variasi input. Pengujian perbandingan menggunakan algoritma penelitian terdahulu menunjukkan empat fungsi umpan balik yang dipilih menghasilkan bilangan yang acak. Uji enkripsi menunjukkan plainteks dan cipherteks memiliki tingkat korelasi "sangat rendah" dan "rendah" dimana rata-rata korelasi untuk setiap pengujian plainteks mendekati 0. Pembangkitan kunci LFSR dengan skema algoritma A5/1 menggunakan empat fungsi XOR dapat menghasilkan keluaran acak yang sangat baik untuk digunakan dalam Stream Cipher.

Kata Kunci: *Skema A5/1, Stream Cipher, Linear Feedback Shift Register*

1. Pendahuluan

Kriptografi sering digunakan sebagai Teknik untuk menyembunyikan pesan yang bersifat rahasia agar tidak mudah diketahui oleh pihak lain. Teknik Keacakan merupakan salah satu Teknik kriptografi yang tercatat digunakan untuk menyembunyikan pesan yang ingin di sampaikan denga cara mengacaknya menjadi tidak beraturan. Teknik pengacakan menghasilkan suatu barisan acak yang sulit dipecahkan. Teknik pengacakan ini dapat dilakukan dengan metode LFSR (Linear Feedback Shift Register), dimana metode ini berguna untuk membangkitkan dan menghasilkan barisan acak [1].

LFSR atau *linear feedback shift register* adalah salah satu sungsi pembangkit kunci untuk *stream cipher* yang berfungsi menerima masukkan dan mengembalikan nilai fungsi ke register geser. Hal inilah yang kemudian menjadikan LFSR mampu menghasilkan bit acak dengan periode maksimal yang mudah untuk diaplikasikan dalam berbagai persoalan kriptografi. Bit masukan LFSR juga dapat di bangkitkan menggunakan operasi XOR dari

ISSN: 2720-992X

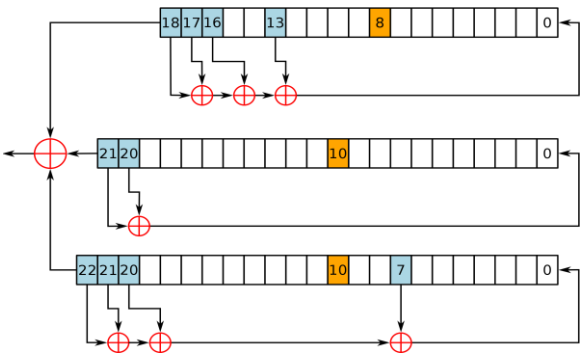
bit keseluruhan register geser atau disebut fungsi umpan balik. Fungsi umpan balik adalah komponen penting yang menunjang luaran bit yang acak [2].

Penelitian ini menggunakan algoritma LFSR dengan skema A5/1 sebagai penguji pembangkit bilangan acak karna skema ini memiliki sifat unik dengan kompleksitas yang berbeda dari skema LFSR lainnya. Peneliti juga memakai empat blok bit XOR untuk mengetahui seberapa baik algoritma membangkitkan bilangan acak dengan membandingkan beberapa pengujian seperti uji enkripsi dan uji visualisasi. Juga digunakan perbandingan pengujian terdahulu untuk mengetahui apakah perbedaan dalam banyaknya blok bit dalam fungsi XOR dapat mempengaruhi bilangan acak yang di hasilkan [3].

2. Metodologi Penelitian

2.1. Algoritma A5/1

A5/1 adalah algoritma *stream cipher* yang digunakan untuk memberikan keamanan privasi dalam komunikasi GSM. Algoritma A5/1 pertama kali diperkenalkan oleh Marc Briceno pada tahun 1999. Algoritma ini memiliki kunci masukkan sebanyak 64 bit sebagai pembangkitnya. Dimana ke-64 bit tersebut terdiri dari tiga LFSR yang berisi 19, 22, dan 23 bit, yang digunakan untuk menghasilkan barisan bit yang acak.

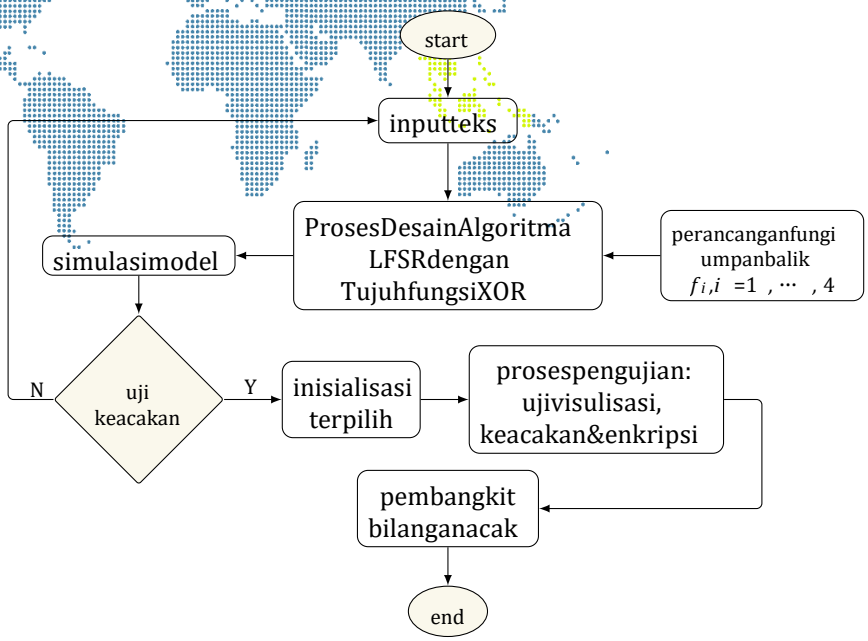


Gambar 1. Skema Algoritma A5/1

Pada Gambar 1 terlihat bahwa LFSR baris pertama (R1) memiliki total 19 bit yang dihitung mulai dari 0, dimana polinomial umpan balik berada pada bit 13, 16, 17, dan 18. Pada baris kedua (R2) yang memiliki total 22 bit, polinomial umpan balik berada pada bit 20, dan 21. Kemudian, pada LFSR baris terakhir (R3) yang memiliki total 23 bit, polinomial umpan balik berada pada bit 7, 20, 21, 22. Pada register R1, R2 dan R3 terdapat bit *clocking*. Dimana pada register R1 ada pada bit ke-8, register R2 di bit ke-10 dan untuk register R3 juga ada di bit ke-10.

2.2. Rancangan Penelitian

Alur kerja penelitian secara umum dijelaskan pada Gambar 3. Penelitian ini menggunakan pengembangan *Linear Feedback Shift Register* (LFSR) pada *Stream cipher* sebagai hasil luaran dari proses penelitian ini. Dimana setiap fungsi pada LFSR akan selalu periodik atau berulang pada $2n– 1$, dengan n adalah banyak bit. Pada awal proses, peneliti membuat bagan pengembangan LFSR pada *Stream cipher* dengan membagi 128 bit menjadi 4 LFSR, dimana masing-masing memiliki besaran bit yang berbeda-beda yaitu 45 bit untuk LFSR pertama, 22 bit LFSR kedua, 33 bit pada LFSR ketiga, dan 28 bit untuk LFSR terakhir.



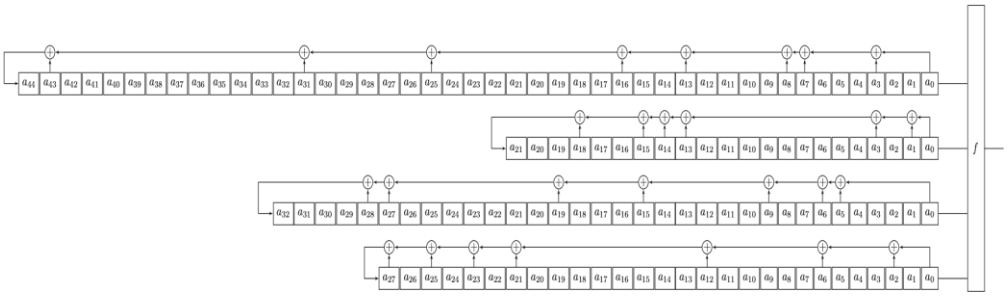
Gambar 2. Skema Umum Proses Penelitian

Sehingga peneliti dapat menentukan bit mana saja yang nantinya akan di-XOR pada setiap baris yang telah dibagi agar mendapatkan hasil LFSR sebagai keluaran. Melakukan konversi terhadap input ke dalam kode ASCII, setelah itu lakukan proses padding untuk merubah kode dari input menjadi bilangan biner (0 dan 1) agar dapat melakukan pencarian terhadap kunci. Hasil dari keluaran pada setiap baris akan kembali di-XOR untuk memperoleh kaluaran terakhir (*f*). Setelah melalui proses XOR pada setiap bit, dilanjutkan dengan pengujian keacakan mengguanakan metode *run test*, *monobit* dan *block bit*. Pengujian LFSR pada *stream cipher* hanya memperoleh keluaran bit tunggal sehingga lebih mudah untuk mencari kunci dan enkripsi dari setiap bit yang dihasilkan oleh input.

3. Hasil dan Pembahasan

3.1. Rancangan Algoritma LFSR

Penenlitian ini merancang algoritma LFSR dengan menggunakan empat fungsi umpan balik, dimana setiap fungsi umpan balik digunakan operasi XOR untuk menentukan nilai bit yang baru pada iterasi berikutnya. Setiap fungsi umpan balik mempunyai komposisi yang berbeda dalam menentukan banyak input yang akan digunakan, dan keempat fungsi umpan balik juga menjadi input untuk mendapatkan hasil pada fungsi utama (*f*). Algoritma LFRS secara umum dibuat dalam bentuk bagan, yang diberikan pada Gambar 4.



Gambar 3. Desain Algoritma A5/1 dengan 4 Fungsi XOR

Setiap fungsi umpan balik dinyatakan dalam f_i, dimana untuk setiap i = {1,2,⋯ ,4} yang dinyatakan pada Persamaan 12. Model diregenrasi menggunakan operasi XOR, dan

setiap iterasi mengunakan fungsi yang sama untuk masing-masing f_i, hasil dari iterasi pertama f_i akan menjadi input pada fungsi utama f.

$$\begin{aligned} f1 &= a_0 \oplus a_3 \oplus a_{13} \oplus a_{16} \oplus a_{25} \oplus a_{31} \oplus a_{43} \\ f2 &= a0 \oplus a_1 \oplus a_3 \oplus a_{13} \oplus a_{14} \oplus a_{15} \oplus a_{18} \\ f3 &= a_0 \oplus a_5 \oplus a_6 \oplus a_9 \oplus a_{15} \oplus a_{19} \oplus a_{27} \oplus a_{28} \\ f4 &= a_0 \oplus a_2 \oplus a_6 \oplus a_{12} \oplus a_{21} \oplus a_{23} \oplus a_{25} \oplus a_{27} \end{aligned} \quad (1)$$

$$f = f_1 \oplus f_2 \oplus f_3 \oplus f_4 \quad (2)$$

Fungsi utama yang menggabungkan setiap fungsi umpan balik f_i diberikan pada Persamaan 2.

3.2. Proses pembangkitan bilangan acak

Nilai inisialisasi dilakukan dengan menentukan batasan bit kunci hanya 16 bit saja, dan mengkonversikan plainteks awal kedalam kode ASCII yang kemudian di ubah menjadi bilangan biner. Dengan mengambil dan menguji beberapa input, Maka akan diperoleh keluaran dari setiap inisialisasi pada LFSR. Setiap keluaran yang dihasilkan akan kembali dilakukan peng-XOR-an untuk mendapat keluaran final. Salah satu plainteks yang di ambil adalah “FTI UK” sehingga mendapat hasil 100 bit keluaran final: 010001100101010
00100100100100000010101010100101100100000001000000010000000010000000010000000100000000100000001000000001000000010000000100000.

3.3. Pengujian Keacakan

Metode pengujian statistik yang digunakan adalah *Run Test*, *Mono Bit* dan *Block Bit*, Pengujian dilakukan menggunakan program yang dibuat melalui *Microsoft Office exce*, digunakan nilai uji keacakan α = 1%, apabila *p-value* < α = 0.01 dinyatakan tidak acak, dan sebaliknya adalah acak. Hasil pengujian dari 15 kunci yang berbeda diberikan pada Tabel 1, Tabel 2, Tabel 3.

Tabel 1. Hasil Pengujian Run Test

No	Data yang Diuji	*p-value*	Hasil Pengujian
1	fti u	0.225425622795732	acak
2	FTI UK	0.559135244489767	acak
3	UKSW FT	0.332956771362852	acak
4	uksw ft	0.067298007332987	acak
5	FTIUKSWBLOTONGAN	0.549898861741019	acak
6	ftiukswblo t ongan	0.686433952209867	acak
7	FTIukswTi	0.064115867295923	acak
8	UKSWftitI	0.936683091627723	acak
9	J@iL0l0	0.853497167231941	acak
10	sAl*TIgA	0.741392633233746	acak
11	0rAn! T!muR	0.332956971362852	acak
12	!3MaranG	0.788703482802642	acak
13	AAAAAAb	0.538369767880286	acak
14	/¡¿3!)	0.673563772819122	acak
15	T I m u R	0.265014721034288	acak
	Rata - rata	0.507696395681383	acak

Tabel 2 adalah hasil dari pengujian menggunakan metode *Run Test*, nilai inputan *p-value* dinyatakan acak karna setiap *p-value* > α = 0.01. Nilai rata - rata dalam pengujian ini adalah 0.507696395681383 dengan inputan terendah adalah inputan ”FTIukswTi” dengan nilai = 0.064115867295923. sedangkan inputan dengan nilai tertinggi iyalah

”UKSWftitI” dengan nilai = 0.936683091627723. dengan hasil *Run Test* yang berbeda dan acak bisa dikatakan bahwa rancangan LFSR ini dapat dijadikan kunci pada kriptografi.

Tabel 2. Hasil Pengujian Mono Bit

No	Data yang Diuji	*p-value*	Hasil Pengujian
1	fti u	0.254945164317321	acak
2	FTI UK	0.254945464317321	acak
3	UKSW FT	0.528189256865538	acak
4	uksw ft	0.375548075920582	acak
5	FTIUKSWBLOTONGAN	0.375920807558254	acak
6	ftiukswblo t ongan	0.370585922548075	acak
7	FTIukswTi	0.704336413488452	acak
8	UKSWftitI	0.899343188561366	acak
9	J@iL0l0	0.254947321516431	acak
10	sAl*TIgA	0.527089655382568	acak
11	0rAn! T!muR	0.527089256865538	acak
12	!3MaranG	0.057779571123597	acak
13	AAAAAAb	0.704336845241348	acak
14	/¡¿3!)	0.899343178561326	acak
15	T I m u R	0.899343161366885	acak
	Rata - rata	0.508916219575643	acak

Tabel 3 adalah hasil pengujian dengan metode *Mono Bit*, *p-value* dari semua data menunjukkan hasil keacakan yang baik. Nilai rata - rata dalam pengujian ini adalah 0.5089162195756 43, Hasil yang konsisten acak dari pengujian *Mono Bit* ini dapat dikatakan bahwa rancangan LFSR ini dapat membangkitkan kunci pada kriptografi.

Tabel 3. Hasil Pengujian Block Bit

No	Data yang Diuji	*p-value*	Hasil Pengujian
1	fti u	0.998316986507179	acak
2	FTI UK	0.999884783439736	acak
3	UKSW FT	0.855824373609523	acak
4	uksw ft	0.999999999999188	acak
5	FTIUKSWBLOTONGAN	0.999999999988210	acak
6	ftiukswblo t ongan	0.988095496143643	acak
7	FTIukswTi	0.916082057968697	acak
8	UKSWftitI	0.999284911554276	acak
9	J@iL0l0	0.996197007938324	acak
10	sAl*TIgA	0.999999329614089	acak
11	0rAn! T!muR	0.999998006027216	acak
12	!3MaranG	0.999999978352155	acak
13	AAAAAAb	0.999998839714638	acak
14	/¡¿3!)	0.855829523437360	acak
15	T I m u R	0.767001898289428	acak
	Rata - rata	0.958434212838911	acak

Hasil pengujian setiap data pada metode *Frequency Test within a Block* secara keseluruhan diperoleh nilai *p-value* > α, sehingga setiap data berada dalam kategori acak.

3.4. Perbandingan Pengujian

Perbandingan pengujian melibatkan tiga penelitian terdahulu yaitu penelitian yang ditulis oleh Aditya Julian Herman sebagai Penelitian 2, penelitian yang di tulis oleh Maria

Nafurbenan Rademta sebagai Penelitian 3, dan penelitian yang ditulis oleh Vondang Manullang Riance Penelitian 4 yang akan dibandingkan dengan algoritma penulis. Penelitian akan mengambil hasil dari pengujian *Runt Test, Block Bit, Mono Bit* dengan kondisi input yang dimasukan adalah sama.

Tabel 4. Hasil Perbandingan Pengujian Run Test

Data yang Diuji	Penelitian 1	Penelitian 2	Penelitian 3	Penelitian 4
fti u	0.22542562	0.79074840	0.67571817	0.17500977
FTI UK	0.55913524	0.80125959	0.84314009	0.33295677
UKSW FT	0.33295677	0.67356377	0.60874248	0.21902269
uksw ft	0.06729800	0.79251807	0.60647856	0.53836976
FTIUKSWBLOTONGAN	0.54989886	0.47131665	0.55913524	0.67356377
ftiukswblo t ongan	0.68643395	0.21120418	0.32695451	0.67356377
FTIukswTi	0.06411586	0.18193426	0.87865030	0.39623540
UKSWftitI	0.93668309	0.53836976	0.73545575	0.06729800
J@iL0l0	0.85349716	0.67356377	0.74727788	0.21120418
sAl*TIgA	0.74139263	0.74139263	0.99738131	0.99235898
0rAn! T!muR	0.33295677	0.16565785	0.26679523	0.46677151
!3MaranG	0.78870348	0.79074840	0.16177069	0.45646078
AAAAAAb	0.53836976	0.47131665	0.27037987	0.95792090
/¡¿3!)	0.67356377	0.60647856	0.53836976	0.06994300
T I m u R	0.26501472	0.99550382	0.27037987	0.73545575
Rata - rata	0.50769638	0.59370509	0.5657753	0.46440900

Tabel 4 perbandingan *Runt Test* yang dilakukan dengan hasil beragam. Ratarata pengujian tiap algoritma menunjukan bahwa pengujian yang dilakukan dapat dikatakan acak, namun terdapat hasil pengujian yang bernilai sama.

Tabel 5. Hasil Perbandingan Pengujian Mono Bit

Data yang Diuji	Penelitian 1	Penelitian 2	Penelitian 3	Penelitian 4
fti u	0.254945	0.8993431	0.7043364	0.2549451
FTI UK	0.254945	0.3759205	0.5270892	0.5270892
UKSW FT	0.528189	0.8993431	0.7043364	0.3759205
uksw ft	0.375548	0.7043364	0.8993431	0.7043364
FTIUKSWBLOTONGAN	0.375920	0.5270892	0.2549451	0.8993431
ftiukswblo t ongan	0.370585	0.7043364	0.8993431	0.8993431
FTIukswTi	0.704336	0.1641035	0.8993431	0.7043364
UKSWftitI	0.899343	0.7043364	0.8993431	0.3759205
J@iL0l0	0.254947	0.8993431	0.3759205	0.7043364
sAl*TIgA	0.527089	0.5270892	0.7043364	0.8993431
0rAn! T!muR	0.527089	0.5270892	0.7043364	0.7043364
!3MaranG	0.057779	0.8993431	0.8993431	0.0577795
AAAAAAb	0.704336	0.5270892	0.5270892	0.5270892
/¡¿3!)	0.899343	0.8993431	0.7043364	0.2549451
T I m u R	0.899343	0.7043364	0.5270892	0.8993431
Rata - rata	0.508916	0.6641628	0.6820354	0.5858938

Pada tabel 5 pengujian dilakukan dengan empat algoritma yang sama dengan pengujian *Mono Bit*. Pengujian keempat algoritma dapat dikatakan acak karena hasil pengujian selalu lebih besar dari nilai α = 1%.

Tabel 6. Hasil Perbandingan Pengujian Block Bit

Data yang diuji	Penelitian 1	Penelitian 2	Penelitian 3	Penelitian 4
fti u	0.9983169	0.3446305	0.5769347	0.9664914
FTI UK	0.9998847	0.9997076	0.9999838	0.8891879
UKSW FT	0.8558243	0.0497870	0.9995406	0.9998847
uksw ft	0.9999918	0.9664914	0.9664914	0.9999561
FTIUKSWBLOTONGAN	0.9999988	0.9999999	0.9999998	0.9999286
ftiukswblo t ongan	0.9880955	0.9995406	0.9373523	0.7108857
FTIukswTi	0.9160820	0.9999965	0.9999999	0.9988975
UKSWftitI	0.9992849	0.9664914	0.8558243	0.9999996
J@iL0l0	0.9961970	0.7108857	0.9961970	0.9829836
sAl*TIgA	0.9999993	0.9999965	0.9999999	0.9999999
0rAn! T!muR	0.9999980	0.9999999	0.9917693	0.9759708
!3MaranG	0.9999999	0.2689124	0.9664914	0.9995406
AAAAAAb	0.9999988	0.9998156	0.9829836	0.9999998
/¡¿3!)	0.8558295	0.9992849	0.7108857	0.9999561
T I m u R	0.7670019	0.9995406	0.9664914	0.9992849
Rata - rata	0.958434	0.8203387	0.9300630	0.9681978

Hasil yang diperoleh pada pengujian *Run Test dan Mono Bit*, terjadi juga pada pengujian dengan *Block Bit*, semua input kunci yang digunakan menghasilkan p-value yang selalu lebih besar dari nilai α = 1% seperti yang diberikan pada tabel 7.

3.5. Pengujian Enkripsi

Pengujian enkripsi dan dekripsi menjadi bagian yang penting untuk dilakukan, sehingga dapat memastikan apakah algoritma rancangan LFSR yang dirancang dapat digunakan sebagai kunci dalam kriptografi. Digunakan dua pengujian, pertama adalah pengujian visulisasi dan kedua adalah pengujian variasi kunci.

$$E_k : P \oplus K = C, \qquad D_k : C \oplus K = P \tag{3}$$

Untuk pengujian enkrispi (E_k) dan dekripsi (D_k), digunakan model sederhana yang operasinya berbasis pada proses XOR seperti yang diberikan pada Persamaan 3. Digunakan (P) plainteks, (C) sebagai cipherteks, dan (K) adalah kunci.

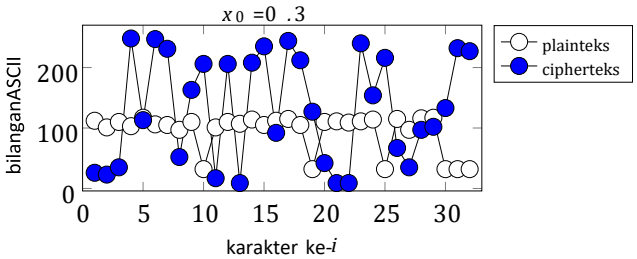
Tabel 7. Hasil Uji Enkripsi

Input Kunci	Plainteks yang Diuji				Tingkat Relasi
	Plainteks 1	Plainteks 2	Plainteks 3	Rata-Rata	
fti u	-0.01720	0.07830	-0.04329	0.00593	sangat kecil
FTI UK	-0.14115	-0.20526	0.14997	-0.06548	sangat kecil
UKSW FT	-0.08116	0.10343	-0.00361	0.00621	sangat kecil
uksw ft	-0.20805	-0.28770	0.26575	-0.07666	sangat kecil
FTIUKSWBLOTONGAN	0.02307	-0.02409	-0.06240	-0.02114	sangat kecil
ftiukswblo t ongan	0.24643	0.07136	-0.32775	-0.00338	sangat kecil
FTIukswTi	-0.21525	0.02195	0.16158	-0.01057	sangat kecil
UKSWftitI	0.03819	0.06369	-0.31634	-0.07148	sangat kecil
J@iL0l0	-0.15482	-0.27630	-0.20282	-0.21131	sangat kecil
sAl*TIgA	-0.24002	0.22725	-0.23183	-0.08154	sangat kecil
0rAn! T!muR	-0.31293	0.13995	-0.19348	-0.12215	sangat kecil
!3MaranG	0.12182	0.17120	0.29811	0.19704	rendah
AAAAAAb	0.35350	-0.15816	0.23685	0.14406	rendah
/¡¿3!)	-0.18075	-0.30704	0.10385	-0.12799	sangat kecil
T I m u R	-0.05259	0.06499	-0.17227	-0.05329	sangat kecil

Terdapat hasil yang beragam pada setiap pengujian dengan rata-rata korelasi yang diperoleh mendekati nol. Hasil ini menunjukkan bahwa rancangan algoritma LFSR dengan empat fungsi XOR dapat menyembunyikan informasi penting dari plainteks, sehingga pada cipherteks menjadi tidak nampak walaupun terdapat hasil rata-rata pengujian pada inputan ”l3MaranG” dengan nilai rata-rata 0.197046372 dan ”AAAAAAb” dengan nilai rata-rata 0.144064731 melebihi nol sehingga kriteria yang dihasilkan adalah rendah. Dengan demikian rancangan algoritma berhasil menjadi pembangkit bilangan acak berbasis LFSR.

3.6. Pengujian Visualisasi

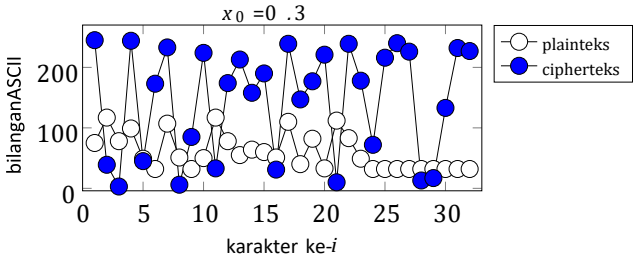
Pengujian selanjutnya visualisasi yang melihat perbedaan atanta plainteks dan cipherteks menggunakan diagram garis. Tiga plainteks pada pengujian sebelumnya digunakan kembali, dan dipilih juga ”FTIUKSWBLOTONGAN” sebagai kunci. Berikut adalah hasil pengujian.



Gambar 4. Perbandingan Plainteks-1 dan Cipherteks

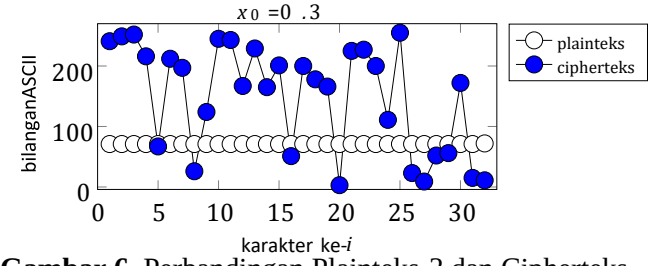
Gambar 4 adalah hasil pengujian untuk plainteks 1, pengujian enkripsi nomor satu. plainteks berada pada interval data 0 ≤ plainteks ≤ 118, sedangkan cipherteks memiliki interval yang lebih besar 0 ≤ cipherteks ≤ 244.

Penggunaan plainteks s4L@t1gA H@t1 b3RIm4N ditunjukkan pada Gambar 6. Nampak ruang plainteks yang lebih kecil dibandingkan dengan plainteks 1, 0 ≤ plainteks ≤ 207 dan memiliki interval yang besar juga, 0 ≤ cipherteks ≤ 126.



Gambar 5. Perbandingan Plainteks-2 dan Cipherteks

Pengujian yang terakhir adalah pengujian ekstrim dimana menggunakan plainteks yang mempunyai karakter yang sama, SSSSSSSSSSSSSSSSSSSSSSSSSSSSST. Hasil yang diperoleh Interval plainteks dalam 83 ≤ plainteks ≤ 72, sedangkan cipherteks memiliki interval 0 ≤ cipherteks ≤ 198.



Gambar 6. Perbandingan Plainteks-3 dan Cipherteks

Pengujian visualisasi grafik antara plainteks dan cipherteks menunjukkan, kunci yang diperoleh dari rancangan LFSR dengan empat fungsi umpan balik berhasil menghasilkan cipherteks yang selalu stabil pada interval yang besar ($0 \leq$ cipherteks ≤ 256).

Suatu usaha untuk menciptakan algoritma yang optimum terhadap kompleksitas waktu dan ruang, adalah keinginan dari para kriptografer. Secara teori, kriptografi *One Time Pad* (OTP) adalah algoritma yang tidak dapat dipecahkan. Penelitian ini merancang LFSR dengan empat fungsi umpan balik sebagai pembangkit bilangan acak, adalah sebuah proses untuk menuju pada konsep OTP, sehingga dapat menghasilkan algoritma yang kompleksitas waktu dan ruang yang baik.

4. Kesimpulan

Pada penelitian ini menunjukkan bahwa algoritma A5/1 sangat baik membangkitkan bilangan acak dengan tingkat korelasi yang baik dalam *Stream Cipher*. Pengujian keacakan menggunakan metode *Run Test, Mono Bit* dan *Block Bit* dengan program yang dibuat menggunakan *Microsoft Office exce* ini selalu menghasilkan *p-value* kurang dari α = 1%, sehingga setiap kunci yang diuji memiliki keluaran acak yang sangat baik. Pengujian dengan menggunakan tiga algoritma yang berbeda juga menghasilkan keluaran dengan keacakan yang baik. Uji enkripsi yang dilakukan dengan tiga kunci yang berbeda menujnukkan bahwa korelasi antara plainteks dan cipherteks baik dengan tingkat korelasi "sangat rendah" pada mayoritas input dan "rendah " pada dua input yang diuji. Dapat disimpulkan bahwa Pembangkit Kunci LFSR dengan Empat Blok Fungsi XOR dapat menghasilkan korelasi kunci yang baik dan keluaran acak yang sangat aman untuk digunakan dalam sebuah *Stream Cipher*.

Daftar Pustaka

[1] Herman, Aditya Julian, *Desain Pembangkit Kunci LFSR dengan Skema A5/1 Menggunakan 7 Blok Bit Fungsi XOR*. Skripsi S-1 Teknik Informatika, FTI UKSW, 2023.

[2] Rademta, Maria Nafurbenan, *Perancangan Enam Bagan Fungsi XOR sebagai Umpan Balik sebagai Pembangkit Bilangan Acak LFSR dengan Skema A5/1*. Skripsi S-1 Teknik Informatika, FTI UKSW, 2023.

[3] Riance, Vondang Manullang, *Desain Lima Fungsi Umpan Balik LFSR dengan Skema A5/1 sebagai Pembangkit Kunci Kriptografi Stream Cipher*. Skripsi S-1 Teknik Informatika, FTI UKSW, 2023.

[4] Federal Information Processing Standards Publication, *Data Encryption Standard (DES)*, U.S. Department of Commerce: National Institute of Standards and Technology (NIST), 1979.

[5] Federal Information Processing Standards Publication, *Advanced Encryption Standard (DES)*, U.S. Department of Commerce: National Institute of Standards and Technology (NIST), November 2001.

[6] Biryukov, A. & De Canni`ere, C., "Data Encryption Standard (DES)", *IBM Journal of Research and Development*, pp. 243-250, Springer: 2011.

[7] Daemen, J. & Rijmen, V., *The Design of Rijndael: AES The Advanced Encryption Standard*, Springer-Verlag, 2001.

[8] Sol´ıs-Sa´nchez, H., & Barrantes E.G., *Using the Logistic Coupled Map for Public Key Cryptography under a Distributed Dynamics Encryption Scheme*, Information, Vol 160, 2018.

[9] Wowor, A.D., et. all, *Domain Examination of Chaos Logistich Function As A Key Operator in Cryptography*. International Journal of Electrical and Computer Engineering, Vol. 8, No. 6, pp. 4577-4583, Institute of Advanced Engineering and Science, 2018

[10] Wowor, Alz Danny, *Regenerasi Fungsi Polinomial Dalam Rancangan Algoritma Berbasis CSPRNG Chaos Sebagai Pembangkit Kunci pada Kriptografi Block Cipher*. Limits Journal, 14 : 4, 2017.
[11] Yopeng, M. R, & Wowor, A. D., *The Implementation of* $f(x) = 3(x^3 - x^2 - x) + 2$ *as CSPRNG Chaos-Based Random Number Generator*. Indonesian Journal on Computing (Indo-JC), 6(1), 41-52.
[12] Chapra, S.C., & Canale, R.P., *Numerical Methods for Engineers*, Sixth Edition, New York: McGraw-Hill, 2010.
[13] Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, A., Dray, J., & Vo, S., *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, Gaithersburg: National Institute of Standards and Technology, 2010.
[14] Munir, R., *Pembangkit Bilangan Acak Semu*. STEI ITB: Informatika, 2018.