JIKO (JURNAL INFORMATIKA DAN KOMPUTER)
Februari 2024, Volume: **8**, No. **1** | Pages 161–173
doi: 10.26798/jiko.v8i1.1116
e-ISSN : 2477-3964 – p-ISSN : 2477-4413

ARTICLE

Modifkasi LFSR Dengan Skema A5/1 Dengan Variasi Fungsi XOR Pada Fungsi Umpan Balik

Modification of LFSR with A51/1 Scheme with Variations of the XOR Function in the Feedback Function

Ahmad Ramadhani[1] dan Alz Danny Wowor[*,2]

[1]Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana, Salatiga, Jawa Tengah, Indonesia
[2]Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana, Salatiga, Jawa Tengah, Indonesia
*Penulis Korespondensi: alzdanny.wowor@uksw.edu

(Disubmit 23-10-03; Diterima 23-11-21; Dipublikasikan online pada 24-02-05)

Abstrak

Penelitian ini merancang pembangkit kunci LFSR (*Linear Feedback Shift Register*) dengan skema A5/1 dan tiga fungsi umpan balik. Fungsi asli pada skema A5/1 berturut hanya menggunakan emat, dua, dan empat pada ketiga fungsi umpan baik, hal ini mengurangi kompleksitas operasi fungsi XOR. Modifikasi dilakukan dengan menambah variabel fungsi pada setiap fungsi umpan balik sehingga dapat meningkatkan nilai keacakan dari setiap urutan bit yang dihasilkan. Pengujian keacakan *Run Test, Mono Bit*, dan *Block Bit* digunakan untuk melihat kemampuan rancangan algoritma dalam menghasilakan urutan bit. Hasil pengujian diperoleh bahwa algoritma secara konsisten mengasilkan luaran yang acak dengan berbagai variasi input (dan lebih baik dari skema A5/1 standart. Setiap luaran dari rancangan algoritma dijadikan sebagai kunci dalam pengujian enkripsi, dan diperoleh luaran bit secara konsisten dapat membuat plainteks dan cipherteks tidak memiliki memiliki nilai korelasi yang sangat rendah atau mendekati nol, sehingga rancangan algoritma LFSR sangat baik dan dapat digunakan sebagai pembangkit kunci dalam kriptografi dan pengamanan informasi.

Kata kunci: Linear Feedback Shift Register, Kriptografi, Skema A5/1

Abstract

This research designs an LFSR (*Linear Feedback Shift Register*) key generator with an A5/1 scheme and three feedback functions. The original function in the A5/1 scheme uses only four, two, and four in the three good feed functions, this reduces the complexity of the XOR function operation. Modifications are carried out by adding function variables to each feedback function so as to increase the randomness value of each sequence of bits produced. Randomness testing *Run Test, Mono Bit*, and *Block Bit* is used to see the ability of the algorithm design to produce a sequence of bits. The test results show that the algorithm consistently produces random output with various input variations (and is better than the standard A5/1 scheme. Each output from the algorithm design is used as a key in encryption testing, and the output bits consistently obtained can make the plaintext and ciphertext not has a correlation value that is very low or close to zero, so the design of the LFSR algorithm is very good and can be used as a key generator in cryptography and information security.

KeyWords: Linear Feedback Shift Register, Cryptography, Scheme A5/1

How to Cite: A. Ramadhani *et al.*, "MODIFKASI LFSR DENGAN SKEMA A5/1 DENGAN VARIASI FUNGSI XOR PADA FUNGSI UMPAN BALIK ", *JIKO (JURNAL INFORMATIKA DAN KOMPUTER)*, Volume: **8**, No.**1**, Pages 161–173, Februari 2024, doi: 10.26798/jiko.v8i1.1116.

1. Pendahuluan

Keamanan informasi merupakan sebuah hal yang sangat penting dalam dunia teknologi saat ini, karena digunakan untuk menjaga kerahasiaan data dari entitas yang tidak berkepentingan. Algortima yang sering digunakan adalah untuk mengamankan data adalah kriptografi, dengan menggunakan kunci rahasia untuk melakukan proses enkripsi maupun dekripsi. Kunci memainkan peran yang sangat vital, dan menjadi salah satu barometer dari kekuatan algoritma tersebut. Oleh kerena itu, diperlukan proses pembangkitan kunci yang baik sehingga dapat mengahasilkan luaran yang acak. Sehingga algoritma dapat menghasilkan cipherteks yang optimum, dan mampun untuk menyembunyikan informasi dari plainteks.

Salah satu metode yang yang dapat digunakan untuk membangkitan kunci adalah *Linear Feedback Shift Register* (LFSR) dengan skema A5/1, algoritma ini dapat menghasilkan sikuens bit biner, berupa barisan acak yang susah dipecahkan [cari acuan]. Skema ini menggunakan operasi dari tiga blok fungsi umpan balik ($P \oplus Q \oplus R$) dengan operasi fungsi XOR yang berbeda, seperti yang ditunjukkan pada Persamaan 1.

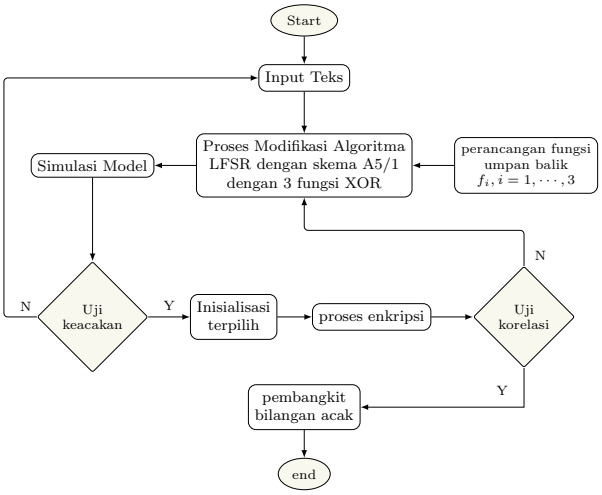
$$\begin{aligned} P &= p_{18} \oplus p_{17} \oplus p_{16} \oplus p_{13} \\ Q &= q_{21} \oplus q_{20} \\ R &= r_{22} \oplus r_{21} \oplus r_{20} \oplus r_7 \end{aligned} \tag{1}$$

dimana $P = \{p_0, p_2, \cdots, p_{18}\}$, $Q = \{q_0, q_2, \cdots, q_{21}\}$, dan $R = \{r_0, r_2, \cdots, r_{22}\}$, sehingga skema A5/1 berukuran 64 bit. Masalah pada skema A5/1 adalah hanya menggunakan operasi fungsi yang sedikit, pada blok-P dengan 4-bit, blok-Q hanya 2-bit dan blok-R dengan 4-bit. Tentunya ini akan mengurangi kompleksitas fungsi dalam operasi proses pembangkitan kunci. Terdapat beberapa penelitian sebelumnya yang mencoba untuk melakukan modifikasi dengan menambah ukuran blok bit dan juga menambah fungsi XOR pada setiap blok umpan balik. Penelitian Herman [1] telah melakukan perancangan kunci menggunakan tujuh blok fungsi XOR, Penelitian [2] dengan enam blok fungsi, Penelitian Manullang [3] dengan lima blok fungsi dan juga [4] dengan empat fungsi. Hasil yang diperoleh dari setiap Penelitian [1], [2],[3] dan [4] dapat menghasilkan luaran bit yang acak berhasil dalam pengujian enkripsi-dekripsi dapat digunakan sebagai kunci dalam kriptografi.

Penelitian ini melakukan modifikasi algoritma A5/1 terhadap fungsi umpan balik, dengan tidak menambah ukuran blok fungsi umpan balik. Melalui penambahan variabel fungsi pada setiap fungsi umpan balik, penelitian ini bertujuan untuk meningkatkan tingkat kompleksitas fungsi dan juga melihat seberapa baik dari urutan bit yang dihasilkan. Untuk mengetahui seberapa baik algoritma yang dirancang dilakukan pengujian keacakan dengan tiga metode yaitu *run test, mono bit,* dan *block bit.* Uji grafik menggunakan scatter plot untuk melihat visualisasi luaran bit dan yang terakhir adalah berupa pengujian enkripsi-denkripsi sehingga dapat mengetahui seberapa baik luaran bit jika dijadikan kunci dalam kriptografi.

2. Rancangan Penelitian

Alur penelitian ini secara garis besar dijelaskan pada Gambar 1. Dalam input teks digunakan tiga jenis kombinasi teks yang kemungkinan besar digunakan oleh setiap *user*. Kombinasi pertama berupa kombinasi huruf, kombinasi kedua berupa kombinasi huruf, angka, dan simbol, dan kombinasi yang ketiga berupa kombinasi huruf yang sama. Pada proses modifikasi algoritma dilakukan perancangan fungsi umpan balik dengan skema A5/1 dengan tiga blok fungsi.



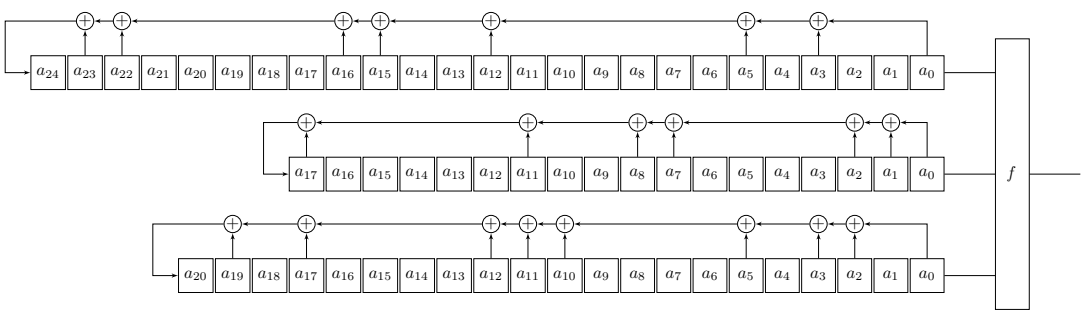
Gambar 1. Skema Proses Penelitian

Dalam melakukan pengujian algoritma yang dimodifikasi dilakukan simulasi model algoritma, simulasi dilakukan dengan mencoba beberapa input teks dan menghasilkan luaran yang acak. Pengujian keacakan sendiri dilakukan dengan menggunakan tiga metode pengujian, metode yang digunakan yaitu *run test, mono bit,* dan *block bit*. Dengan melakukan pengujian dengan beberapa input akan diperoleh luaran dari inisialisasi pada LFSR. Untuk mendapatkan inisialisasi terpilih dilakukan dengan menentukan batasan bit kunci hanya dengan 8 bit dan mengkonversi ke kode *ASCII* dan kemudian diubah menjadi bilangan biner. Inisialisasi tepilih pada LFSR selanjutnya digunakan untuk melakukan pengujian enkripsi untuk memastikan algoritma yang dimodifikasi dapat digunakan sebagai pembangkit kunci. Dalam uji korelasi dilakukan untuk mengetahui perbedaan pola yang tebentuk antara plainteks dan cipherteks berdasarkan interval yang didapat dari setiap plainteks dan cipherteks. Ketika pola yang terbentuk memiliki perbedaan yang signifikan maka algoritma yang dimodifikasi dapat digunakan sebagai pembangkit kunci bilangan acak.

3. Hasil dan Pembahasan

3.1 Rancangan Algoritma LFSR

Penelitian ini merancang algoritma LFSR dengan menggunakan tiga 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 ketiga fungsi umpan balik juga menjadi input untuk mendapatkan hasil pada fungsi utama (*f*). Setiap fungsi umpan balik dinyatakan dalam f_i. Model diregenerasi menggunakan operasi XOR, dan setiap iterasi menggunakan fungsi yang sama untuk masing-masing f_i, hasil dari iterasi pertama f_i akan menjadi input pada fungsi utama *f*. Algoritma LFSR secara umum dibuat dalam bentuk bagan yang diberikan pada Gambar 2.



Gambar 2. Desain Algoritma A5/1 dengan Tiga Fungsi XOR

Pada penelitian ini menggunakan pengembangan LFSR sebagai hasil luaran dari proses penelitian ini. Setiap fungsi LFSR akan selalu berulang pada $2n - 1$, dimana n adalah banyaknya bit. Tahap pertama peneliti membuat bagan pengembangan LFSR dengan membagi 64 bit menjadi 3 LFSR, dengan masing-masing memiliki bit yang berbeda-beda. Yang pertama terdapat 25 bit pada LFSR yang pertama, 18 bit pada LFSR yang kedua, dan 21 bit pada LFSR yang ketiga. Setiap fungsi umpan balik dinyatakan dalam f_i, dimana setiap $i = \{1, 2, 3\}$ yang dinyatakan pada Persamaan 2.

$$\begin{aligned} f_1 &= a_0 \oplus a_3 \oplus a_5 \oplus a_{12} \oplus a_{15} \oplus a_{16} \oplus a_{22} \oplus a_{23} \\ f_2 &= a_0 \oplus a_1 \oplus a_2 \oplus a_7 \oplus a_8 \oplus a_{11} \oplus a_{17} \\ f_3 &= a_0 \oplus a_2 \oplus a_3 \oplus a_5 \oplus a_{10} \oplus a_{11} \oplus a_{12} \oplus a_{17} \oplus a_{19} \end{aligned} \tag{2}$$

$$f = f_1 \oplus f_2 \oplus f_3 \tag{3}$$

Fungsi utama yang menggabungkan setiap fungsi umpan balik f_i diberikan pada Persamaan 3.

3.2 Proses Pembangkitan Bilangan Acak

Nilai inisialisasi dilakukan dengan menentukan batasan bit kunci hanya dengan 8 bit dan mengkonversikan plainteks kedalam kode *ASCII* yang kemudian diubah menjadi bilangan biner. Dengan melakukan pengujian dengan beberapa input maka akan diperoleh keluaran dari setiap inisialisasi pada LFSR. Setiap keluaran yang diperoleh akan kembali dilakukan proses XOR untuk mendapatkan keluaran final. Salah satu input yang diambil adalah "ukswfti" dan mendapat hasil 200 bit keluaran final : 110100111010 10011011101101100011101010010 111011101 000110000 11011010000110100000101011110000100010001010 00000010110111010010010010000100100110111000100010010111011101110111111110111101111001101101 00110.

3.3 Pengujian Keacakan

3.3.1 Pengujian Run Test

Metode pengujian statistik yang pertama menggunakan metode *Run Test*. Pengujian dilakukan dengan mengggunakan program yang dibuat pada *Microsoft Office Excel*. Nilai uji keacakan yang digunakan yaitu $\alpha = 1\%$, apabila *p-value* $< \alpha = 0.01$ maka dinyatakan tidak acak, dan sebaliknya maka acak. Hasil pengujian dengan metode *Run Test* dari 30 kunci yang berbeda secara berturut-turut diberikan pada Tabel 1 Tabel 2 dan Tabel 3.

Tabel 1. Hasil Uji Run Test

No	Input teks	*p-value*	Hasil Pengujian
1	ahmad	0,989128372	acak
2	rama	0,160489951	acak
3	dhani	0,56542073	acak
4	fti	0,335296572	acak
5	uksw	0,242976381	acak
6	ukswfti	0,762648718	acak
7	progdi	0,666275713	acak
8	sarjana	0,76494606	acak
9	teknik	0,772696659	acak
10	informatika	0,471036489	acak
	Rata-rata	0,573091565	

Pada Tabel 1 menunjukkan hasil pengujian dengan metode *run test* yang dimana menggunakan 10 kunci dengan jenis tulisan biasa yang terdiri dari kombinasi huruf. Hasil yang diperoleh dari pengujian 10 kunci kombinasi huruf didapat semua hasil acak karena nilai *p-value* lebih besar dari 0.01. Nilai rata-rata pada 10 kunci dengan kombinasi huruf yaitu 0,573091565 dengan nilai *p-value* tertinggi didapat dari kunci "ahmad" dengan nilai 0,989128372.

Tabel 2. Hasil Uji Run Test

No	Input teks	*p-value*	Hasil Pengujian
1	4hm4d	0,101892421	acak
2	r4m4	0,840986065	acak
3	dh4n!	0,396933762	acak
4	4!4mat	0,176822365	acak
5	Sal4t194	0,132783344	acak
6	t3g4lr3j0	0,97642178	acak
7	4r90muly0	0,673190455	acak
8	j4w4	0,287171528	acak
9	t3n94h	0,471036489	acak
10	1nd0n3514	0,065013406	acak
	Rata-rata	0,412225162	

Pada Tabel 2 menunjukkan hasil pengujian dengan metode *run test* yang dimana menggunakan 10 kunci dengan kombinasi huruf, angka, dan simbol. Hasil yang diperoleh *p-value* dari inputan dinyatakan acak karena setiap *p-value* > α = 0.01. Nilai rata-rata pada 10 kunci dengan kombinasi huruf, angka, dan simbol yaitu 0,412225162 dengan nilai *p-value* terendah didapat dari kunci "1nd0n3514" dengan nilai *p-value* 0,065013406.

Tabel 3. Hasil Uji Run Test

No	Input teks	*p-value*	Hasil Pengujian
1	ZZZZZY	0,254550257	acak
2	BCCCC	0,164898182	acak
3	GGGGGGH	0,407740815	acak
4	RQQQQQ	0,746049304	acak
5	SSSSST	0,509063597	acak
6	DEEEEEE	0,683711926	acak
7	KKKKKL	0,249792767	acak
8	MNNNNN	0,067136578	acak
9	JJJJJJI	0,956153434	acak
10	UVVVVVV	0,683711926	acak
	Rata-rata	0,472280879	

Pada Tabel 3 menunjukkan hasil pengujian 10 kunci dengan kombinasi huruf yang sama. Hasil pengujian mendapatkan semua hasil acak dengan nilai *p-value* lebih besar dari 0.01 dengan nilai rata-rata yang didapat yaitu 0,472280879. Dengan hasil pengujian 30 kunci berbeda dengan metode *run test* mendapat hasil acak dan bisa disimpulkan bahwa rancangan LFSR yang dibuat dapat dijadikan kunci pada kriptografi.

3.3.2 Pengujian Mono Bit

Pada pengujian kedua digunakan metode *Mono Bit* yang juga dilakukan menggunakan program yang dibuat pada *Microsoft Office Excel*. Dalam pengujian dengan metode *Mono bit* mendapat hasil yang dapat dilihat pada Tabel 4, Tabel 5, dan Tabel 6.

Tabel 4. Hasil Uji Mono Bit

No	Input teks	*p-value*	Hasil Pengujian
1	ahmad	0,423710797	acak
2	rama	1	acak
3	dhani	0,423710797	acak
4	fti	1	acak
5	uksw	0,548506236	acak
6	ukswfti	0,689156517	acak
7	progdi	0,045500264	acak
8	sarjana	0,548506236	acak
9	teknik	0,027806895	acak
10	informatika	0,161513318	acak
	Rata-rata	0,486841106	

Pada Tabel 4 menunjukkan hasil pengujian *mono bit* dengan 10 kunci dengan tulisan biasa yang terdiri dari kombinasi huruf mendapatkan hasil yang diperoleh bahwa nilai *p-value* > α = 0.01 sehingga dapat dinyatakan acak. Nilai rata-rata pada 10 kunci dengan kombinasi huruf yaitu 0,486841106 dengan nilai terendah didapat oleh kunci "teknik" dengan *p-value* 0,027806895.

Tabel 5. Hasil Uji Mono Bit

No	Input teks	*p-value*	Hasil Pengujian
1	4hm4d	0,689156517	acak
2	r4m4	0,689156517	acak
3	dh4n!	0,423710797	acak
4	4!4mat	0,161513318	acak
5	Sal4t194	0,423710797	acak
6	t3g4lr3j0	1	acak
7	4r90muly0	0,423710797	acak
8	j4w4	0,689156517	acak
9	t3n94h	0,161513318	acak
10	1nd0n3514	0,23013934	acak
	Rata-rata	0,489176792	

Pada Tabel 5 menunjukkan hasil pengujian *mono bit* dengan 10 kunci dengan tulisan biasa yang terdiri dari kombinasi huruf, angka, dan simbol. Hasil yang diperoleh *p-value* dari inputan dinyatakan acak karena setiap *p-value* bernilai lebih besar dari 0.01. Nilai rata-rata pada 10 kunci dengan kombinasi huruf yaitu 0,489176792 dengan didapat *p-value* tertinggi yaitu 1 yang merupakan nilai dari kunci "t3g4lr3j0".

Tabel 6. Hasil Uji Mono Bit

No	Input teks	*p-value*	Hasil Pengujian
1	ZZZZZY	0,23013934	acak
2	BCCCC	0,027806895	acak
3	GGGGGGH	0,23013934	acak
4	RQQQQQ	0,109598583	acak
5	SSSSST	0,423710797	acak
6	DEEEEEE	0,23013934	acak
7	KKKKKL	0,317310508	acak
8	MNNNNN	0,161513318	acak
9	JJJJJJI	0,689156517	acak
10	UVVVVVV	0,23013924	acak
	Rata-rata	0,264965388	

Pada Tabel 6 menunjukkan hasil pengujian 10 kunci dengan kombinasi huruf yang sama. Nilai rata-rata dari kunci dengan kombinasi huruf yang sama yaitu 0,264965388. Dengan hasil *mono bit* yang memiliki hasil konsisten acak setiap kunci bisa disimpulkan bahwa rancangan LFSR yang dibuat dapat dijadikan kunci pada kriptografi.

3.3.3 Pengujian Block Bit

Proses pengujian keacakan yang terakhir menggunakan metode *Block Bit*. Pada pengujian dengan metode ini dengan 30 kunci berbeda menghasilkan rata-rata nilai *p-value* hampir mendekati 1. Hasil dari 30 kunci berbeda dapat dilihat pada Tabel 7, Tabel 8 dan Tabel 9.

Tabel 7. Hasil Uji Block Bit

No	Input teks	*p-value*	Hasil Pengujian
1	ahmad	1	acak
2	rama	0,999999876	acak
3	dhani	1	acak
4	fti	0,999983651	acak
5	uksw	1	acak
6	ukswfti	1	acak
7	progdi	1	acak
8	sarjana	1	acak
9	teknik	1	acak
10	informatika	1	acak
	Rata-rata	0,999998353	

Pada Tabel 7 menunjukkan hasil pengujian *block bit* dengan 10 kunci dengan tulisan biasa yang terdiri dari kombinasi huruf. Hasil yang diperoleh menyatakan semua kunci acak karena diperoleh *p-value* bernilai lebih besar dari 0.01. Nilai rata-rata pada 10 kunci dengan kombinasi huruf yaitu 0,999998353 dengan didapat juga nilai terendah pada kunci "fti" dengan nilai *p-value* 0,999983651.

Tabel 8. Hasil Uji Block Bit

No	Input teks	*p-value*	Hasil Pengujian
1	4hm4d	1	acak
2	r4m4	1	acak
3	dh4n!	1	acak
4	4!4mat	1	acak
5	Sal4t194	1	acak
6	t3g4lr3j0	1	acak
7	4r90muly0	1	acak
8	j4w4	1	acak
9	t3n94h	1	acak
10	1nd0n3514	0,999983851	acak
	Rata-rata	0,999998385	

Pada Tabel 8 menunjukkan hasil pengujian *block bit* dengan 10 kunci dengan tulisan biasa yang terdiri dari kombinasi huruf, angka, dan simbol. Hasil yang diperoleh *p-value* dari inputan dinyatakan acak karena setiap *p-value* > α = 0.01. Nilai rata-rata pada 10 kunci dengan kombinasi huruf yaitu 0,999998385 dengan nilai tertinggi yang diperoleh yaitu 1.

Tabel 9. Hasil Uji Block Bit

No	Input teks	*p-value*	Hasil Pengujian
1	ZZZZZY	1	acak
2	BCCCC	1	acak
3	GGGGGGH	1	acak
4	RQQQQQ	1	acak
5	SSSSST	1	acak
6	DEEEEEE	1	acak
7	KKKKKL	1	acak
8	MNNNNN	1	acak
9	JJJJJJI	1	acak
10	UVVVVVV	1	acak
	Rata-rata	1	

Pada Tabel 9 dalam pengujian dengan *block bit* di dapat hasil rata-rata 1 dengan mayoritas mendapat nilai *p-value* 1. Dengan begitu setiap inputan dapat disimpulkan dalam kategori acak dan hasil pengujian dapat dinyatakan rancangan LFSR memenuhi untuk menjadi kunci pembangkit pada kriptografi.

3.4 Perbandingan Pengujian

Pada perbandingan pengujian pada penelitian ini melibatkan penelitian terdahulu yang didalam penelitian tersebut terdapat pengujian dengan tiga penelitian terdahulu. Melibatkan penelitian terdahulu yang berjudul "Desain Pembangkit Kunci LFSR dengan Empat Blok Bit Fungsi XOR Menggunakan Skema A5/1 "[4] yang akan dibandingkan dengan algoritma penulis. Penelitian ini akan menggunakan hasil dari pengujian dengan metode *Run Test, Mono bit*, dan *Block Bit* dengan mengikuti input dari penelitian sebelumnya.

Tabel 10. Perbandingan Run Test

No	Input	Riset 1 [1]	Riset 2 [2]	Riset 3 [3]	Riset 4 [4]	Penelitian 1
1	fti u	0.790748405	0.675718178	0.175009773	0.225425623	0,343364453
2	FTI UK	0.801259597	0.843140093	0.332956771	0.559135244	0,616608073
3	UKSW FT	0.673563773	0.608742484	0.219022696	0.332956771	0,994746847
4	uksw ft	0.792518079	0.606478567	0.538369768	0.067298007	0,768145444
5	FTIUKSWBLOTONGAN	0.471316659	0.559135244	0.673563773	0.549898862	0,009327858
6	ftiukswblo t ongan	0.211204181	0.326954518	0.673563773	0.686433952	0,198961138
7	FTIukswTi	0.181934262	0.878650308	0.396235406	0.064115867	0,245943409
8	UKSWftitI	0.538369768	0.735455754	0.067298007	0.936683092	0,160811529
9	J@iL0l0	0.673563773	0.747277887	0.211204181	0.853497167	0,714987639
10	sAl*TIgA	0.741392633	0.997381318	0.992358982	0.741392633	0,165693555
11	0rAn! T!muR	0.165657851	0.266795235	0.466771518	0.332956771	0,200828164
12	!3MaranG	0.790748405	0.161770699	0.456460784	0.788703483	0,558634646
13	AAAAAAb	0.471316659	0.270379875	0.957920902	0.538369768	0,638620723
14	/¡¿3!)	0.606478567	0.538369768	0.069943008	0.673563773	0,387272532
15	T I m u R	0.995503821	0.270379875	0.735455754	0.265014721	0,571942938
	Rata - Rata	0.593705095	0.56577532	0.464409006	0.507696382	0,438392597

Pada Tabel 10 menunjukan hasil perbandingan pengujian *Run Test* yang beragam dengan input yang sama dengan penelitian sebelumnya. Dengan hasil rata-rata pengujian algoritma penulis mendapatkan hasil acak. Dari hasil pengujian dapat disimpulkan kelima algoritma dapat digunakan kunci dalam kriptografi.

Tabel 11. Perbandingan Mono Bit

No	Input	Riset 1 [1]	Riset 2 [2]	Riset 3 [3]	Riset 4 [4]	Penelitian 1
1	fti u	0.899343189	0.704336413	0.254945164	0.25494516	0,423710797
2	FTI UK	0.375920583	0.527089257	0.527089257	0.25494546	0,548506236
3	UKSW FT	0.899343189	0.704336413	0.375920583	0.52818926	0,689156517
4	uksw ft	0.704336413	0.899343189	0.704336413	0.37554808	0,423710797
5	FTIUKSWBLOTONGAN	0.527089257	0.254945164	0.899343189	0.37592081	0,161513318
6	ftiukswblo t ongan	0.704336413	0.899343189	0.899343189	0.37058592	0,689156517
7	FTIukswTi	0.164103507	0.899343189	0.704336413	0.70433641	0,423710797
8	UKSWftitI	0.704336413	0.899343189	0.375920583	0.89934319	0,841480581
9	J@iL0l0	0.899343189	0.375920583	0.704336413	0.25494732	0,000673859
10	sAl*TIgA	0.527089257	0.704336413	0.899343189	0.52708966	0,423710797
11	0rAn! T!muR	0.527089257	0.704336413	0.704336413	0.52708926	0,548506236
12	!3MaranG	0.899343189	0.899343189	0.057779571	0.05777957	0,689156517
13	AAAAAAb	0.527089257	0.527089257	0.527089257	0.70433685	0,161513318
14	/¡¿3!)	0.899343189	0.704336413	0.254945164	0.89934319	0,045500264
15	T I m u R	0.704336413	0.527089257	0.899343189	0.89934316	0,002699796
	Rata - Rata	0.664162848	0.682035435	0.585893866	0.50891622	0,40484709

Pada Tabel 11 menunjukan hasil perbandingan pengujian *Mono bit* yang beragam dengan input yang sama dengan penelitian sebelumnya. Dari kelima algoritma memiliki hasil rata-rata mendapatkan hasil acak. Pada penelitian 1 memiliki rata-rata terendah dibandingkan dengan empat penelitian sebelumnya, dengan

itu penelitian 1 dapat digunakan dalam pembangkit kunci kriptografi.

Tabel 12. Perbandingan Block Bit

No	Input	Riset 1 [1]	Riset 2 [2]	Riset 3 [3]	Riset 4 [4]	Penelitian 1
1	fti u	0.344630505	0.576934771	0.966491465	0.99831699	1
2	FTI UK	0.999707663	0.999983851	0.889187908	0.99988478	1
3	UKSW FT	0.049787068	0.999540633	0.999884783	0.85582437	1
4	uksw ft	0.966491465	0.966491465	0.999956144	0.99999188	1
5	FTIUKSWBLOTONGAN	0.999999961	0.99999987	0.999928613	0.99999882	1
6	ftiukswblo t ongan	0.999540633	0.937352349	0.71088575	0.98809550	1
7	FTIukswTi	0.999996598	0.99999995	0.998897512	0.91608206	1
8	UKSWftitI	0.966491465	0.855824374	0.999999615	0.99928491	1
9	J@iL0l0	0.71088575	0.996197008	0.98298364	0.99619701	1
10	sAl*TIgA	0.999996598	0.99999996	0.99999998	0.99999933	0,999999615
11	0rAn! T!muR	0.99999999	0.991769347	0.975970864	0.99999801	1
12	!3MaranG	0.268912417	0.966491465	0.999540633	0.99999998	0,999998006
13	AAAAAAb	0.999815653	0.98298364	0.999999876	0.99999884	1
14	/¡¿3!)	0.999284912	0.71088575	0.999956144	0.85582952	1
15	T I m u R	0.999540633	0.966491465	0.999284912	0.76700190	1
	Rata - Rata	0.820338754	0.93006306	0.968197856	0.95843421	0,999999841

Pada Tabel 12 menunjukan hasil perbandingan pengujian *Block Bit* yang beragam dengan input yang sama dengan penelitian sebelumnya. Dari kelima algoritma memiliki hasil rata-rata mendapatkan hasil *p-value* yang selalu lebih besar $\alpha = 1\%$. Sehingga dapat disimpulkan kelima algoritma dapat digunakan dalam membangkitkan kunci kriptografi.

3.5 Ketersediaan Ruang Kunci

Sebuah algoritma yang dirancang dengan menggunakan panjang blok kunci sebesar 64 bit, kriptanalis membutuhkan waktu 2^{64} untuk memecahkan algoritma. Jika sebuah komputer dapat memecahkan 10^6 kunci dalam waktu satu detik, maka waktu yang dibutuhkan untuk membobol seluruh kunci yaitu 593066,6176 tahun. Perancangan algoritma ini juga menggunakan 64 bit dimana algoritmanya dapat menghasilkan beragam cipherteks. Secara umum bahwa algoritma LFSR dengan tiga blok bit fungsi XOR dapat menghasilkan beragam cipherteks yang berbeda sehingga dalam proses pencarian memerlukan waktu 593066,6176 tahun. Dengan begitu tentu pasti mempersulit kriptanalis, meskipun kemungkinan dilakukan seorang kriptanalis masih bisa terjadi. Namun dengan begitu kriptanalis membutuhkan waktu membobol yang lebih lama daripada kriptografi biasa.

3.6 Pengujian Enkripsi-Dekripsi

Dalam memastikan algoritma LFSR yang dirancang dapat digunakan dengan sebagai pembangkit kunci pada kriptografi diperlukan proses pengujian enkripsi dan dekripsi. Digunakan *block cipher* dengan panjang kunci 256 bit (setara 32 karakter), juga untuk domain dan *range* ASCII sehingga operasi berjalan dalam modulus 256.

Dalam pengujian enkripsi digunakan proses enkripsi yang sederhana, dimana menjumlahkan plainteks (P) dan kunci (K), untuk memperoleh cipherteks (C). Dalam pengujian enkripsi digunakan tiga plainteks yang dari setiap plainteks yang digunakan dapat mewakili variasi plainteks yang biasanya digunakan oleh pengguna. Plainteks pertama yaitu ftiuksw blotongan sidorejo salatiga, untuk mewakili tulisan biasa yang hanya terdiri dari kombinasi huruf. Plainteks kedua Pr0gr@m 5tud! t3kN1k !nf012m@tIk4 digunakan kombinasi huruf, simbol, dan angka. Plainteks terakhir adalah TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTI yang mempunyai huruf yang sama.

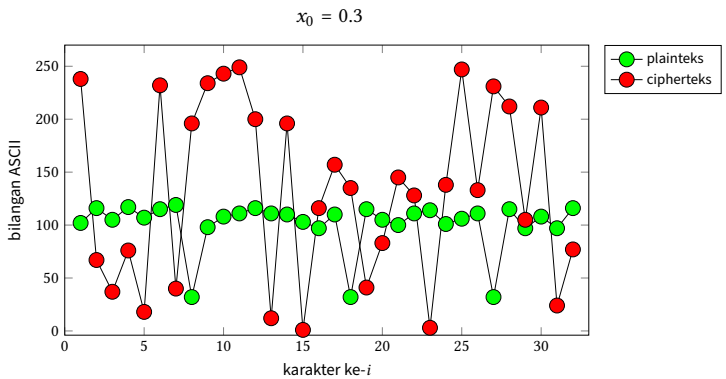
Pengujian korelasi digunakan untuk melihat hubungan antara plainteks dan cipherteks, sehingga dapat diketahui seberapa baik kunci yang digunakan dapat menyamarkan plainteks, sehingga dapat menghilangkan hubungan antara keduanya. Hasil pengujian untuk ketiga palinteks dengan setiap data set diberikan pada Tabel 13.

Tabel 13. Hasil Uji Korelasi

No	Input Kunci	*Plainteks 1*	*Plainteks 2*	*Plainteks 3*	Rata - rata	Nilai Korelasi
1	ahmad	-0,156658484	-0,025176217	-0,185988024	-0,122607575	Sangat Rendah
2	rama	0,278746874	0,210588547	-0,018514132	0,15694043	Sangat Rendah
3	dhani	-0,183086465	-0,002279331	-0,226636535	-0,13733411	Sangat Rendah
4	fti	-0,0259384	-0,078838697	-0,1568309	-0,087202666	Sangat Rendah
5	uksw	-0,155464231	-0,067120648	-0,09263246	-0,105072446	Sangat Rendah
6	ukswfti	0,113947655	-0,0412348	-0,147591112	-0,024959419	Sangat Rendah
7	progdi	0,043264586	-0,103417528	-0,252022969	-0,104058637	Sangat Rendah
8	sarjana	0,010508446	-0,146753088	0,234251329	0,032668896	Sangat Rendah
9	teknik	0,133852291	-0,143537437	-0,065921618	-0,025202255	Sangat Rendah
10	informatika	0,034866067	0,172046669	-0,163331614	0,014527041	Sangat Rendah
11	4hm4d	-0,128951537	-0,217637129	0,22271888	-0,041289929	Sangat Rendah
12	r4m4	0,086761559	0,04249618	0,12284588	0,08403454	Sangat Rendah
13	dh4n!	-0,007087319	0,03820737	0,073768082	0,034962711	Sangat Rendah
14	4!4mat	-0,005603236	0,272594856	-0,033459123	0,077844166	Sangat Rendah
15	Sal4t194	0,010011913	0,241233528	-0,199846298	0,017133048	Sangat Rendah
16	t3g41r3j0	-0,223640154	-0,06042519	0,22606086	-0,019334828	Sangat Rendah
17	4r90mu!y0	-0,016261696	-0,07119616	0,041340813	-0,015372348	Sangat Rendah
18	j4w4	0,243548588	0,046757907	-0,228574032	0,020577488	Sangat Rendah
19	t3n94h	0,257834531	-0,112174798	-0,132716004	0,004314576	Sangat Rendah
20	1nd0n3514	-0,090280228	-0,223181963	-0,345164798	-0,21954233	Sangat Rendah
21	ZZZZZY	-0,099782504	-0,220412724	0,222026704	-0,032722841	Sangat Rendah
22	BCCCC	-0,059362278	0,062427026	-0,252913443	-0,083282898	Sangat Rendah
23	GGGGGGH	0,162732155	0,197872692	0,13073534	0,163780062	Sangat Rendah
24	RQQQQQ	0,110618104	-0,053303798	0,088788263	0,048700856	Sangat Rendah
25	SSSSST	0,324287964	0,193636993	0,023892625	0,180605861	Sangat Rendah
26	DEEEEEE	0,046292177	0,101268701	-0,028441881	0,039706332	Sangat Rendah
27	KKKKKL	0,27187075	-0,147793478	-0,179133596	-0,018352108	Sangat Rendah
28	MNNNNN	0,032033709	-0,169679723	0,283478758	0,048610915	Sangat Rendah
29	JJJJJJI	-0,13213296	0,005075247	-0,030856672	-0,052638128	Sangat Rendah
30	UVVVVVV	-0,269205416	-0,108530731	-0,148396321	-0,175377489	Sangat Rendah

3.7 Pengujian Korelasi

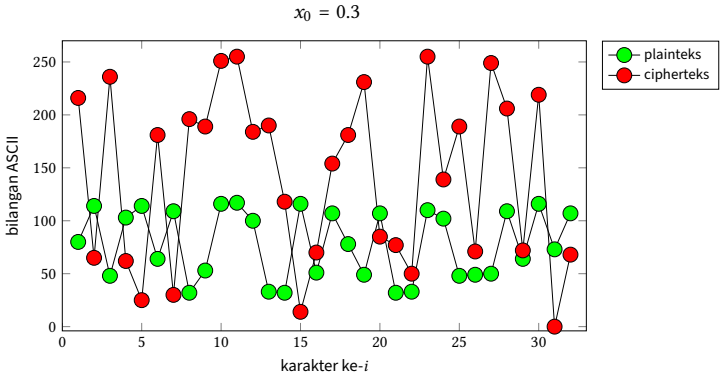
Pengujian korelasi dilakukan untuk mengetahui perbedaan antara plainteks dan cipherteks. Dalam pengujian korelasi digunakan satu kunci dari 30 kunci yang sudah diuji keacakannya. Kunci yang digunakan merupakan kunci yang sekiranya memiliki *p-value* tertinggi yaitu "t3g41r3j0". Untuk melakukan pengujian korelasi digunakan kembali tiga plainteks yang sama dengan yang diuji pada pengujian enkripsi. Pengujian ini digunakan untuk mengetahui apakah kunci yang digunakan dapat membuat pola yang berbeda antara plainteks dan cipherteks. Jika plainteks dan cipherteks mempunyai pola yang berbeda maka kunci dalam algoritma telah berhasil membuat efek difusi dan konfusi pada algoritma.



Gambar 3. *Perbandingan Plainteks 1 dan Cipherteks*

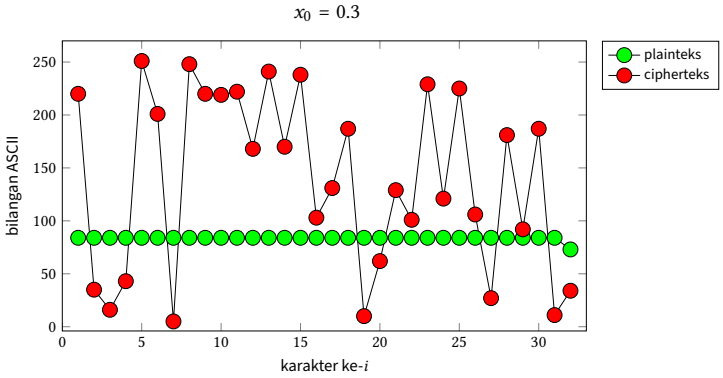
Gambar 3 adalah hasil pengujian untuk plainteks 1, ftiuksw blotongan sidorejo salatiga. Diagram garis plain-

teks dan cipherteks menunjukkan pola yang sangat berbeda. Dimana dapat dilihat pada Gambar 3 bahwa pola yang dibuat dari plainteks dan cipherteks terlihat sangat berbeda dengan plainteks memiliki interval 0 sampai 119 sedangkan interval pada cipherteks lebih tinggi yaitu 0 hingga 249. Sehingga kunci dengan tiga fungsi umpan balik pada algoritma LFSR dapat digunakan dan dapat membuat geometri yang berbeda antara plainteks dan cipherteks.



Gambar 4. *Perbandingan Plainteks 2 dan Cipherteks*

Pada pengujian plainteks kedua dengan plainteks Pr0gr@m 5tud! t3kN1k !nf012m@tIk4 ditunjukan pada Gambar 4 dimana juga menunjukan perbedaan yang signifikan antara pola dari plainteks dan cipherteks. Pada pola plainteks memperoleh interval 0 hingga 117 sedangkan pada pola cipherteks memperoleh hasil interval lebih tinggi yaitu hingga 255. Maka dengan begitu dapat disimpulkan bahwa kunci yang digunakan berfungsi baik dalam melakukan proses enkripsi karena pada diagram garis pada cipherteks dapat menyembunyikan pola yang ada pada plainteks.



Gambar 5. *Perbandingan Plainteks 3 dan Cipherteks*

Pada pengujian yang ketiga merupakan pengujian yang bisa dikatakan ekstrim, karena menggunakan karakter yang sama dalam sebuah plainteksnya. Dengan plainteks TTTTTTTT TTTTTTTTTTTTTTTTTTTTTTTT dapat menghasilkan pola yang bisa dilihat pada Gambar 5 bahwa hasil yang didapat diagram garis plainteks yang membentuk sebuah garis lurus, dapat menghasilkan cipherteks yang fluktuatif. Berdasakan diagram garis pada cipherteks dapat dilihat bahwa kunci dapat menyembunyikan pola yang ada pada plainteks

4. Pembahasan

Pada pengujian enkripsi mendapatkan hasil dengan rata-rata nilai korelasi yang diperoleh mendekati nol. Dengan hasil yang didapat menyimpulkan bahwa rancangan algoritma LFSR dengan tiga fungsi XOR dapat menyembunyikan pesan dari plainteks sehingga pada cipherteks menjadi tidak nampak. Dengan demikian rancangan algoritma yang dibuat berhasil menjadi pembangkit kunci bilangan acak berbasis LFSR.

Pengujian korelasi grafik antara plainteks dan cipherteks menunjukan, kunci yang diperoleh dari pembangkitan bilangan acak berhasil menghasilkan cipherteks yang selalu stabil pada interval yang besar

($0 \leq$ cipherteks ≤ 256). Hasil ini tentunya akan mempersulit penebakan pada plainteks dan cipherteks, karena kunci dapat mempertahankan ruang hasil (*range*) cipherteks. Kriptanalis mungkin akan dapat memecahkan, tetapi tentunya akan memerlukan waktu yang lebih lama walaupun algoritma enkripsi yang digunakan hanya dengan operasi penjumlahan.

Suatu usaha untuk menciptakan algoritma yang optimum terhadap kompleksitas waktu dan ruang, adalah keinginan dari para kriptografer. Penggunaan pembangkit bilangan acak dengan menemukan fungsi yang baru merupakan salah satu cara yang dapat diperhatikan dan dikembangkan. Secara teori, kriptografi *One Time Pad* (OTP) adalah algoritma yang tidak dapat dipecahkan. Penelitian ini merancang suatu algoritma LFSR dengan tiga fungsi XOR, dimana sebuah proses untuk menuju pada konsep OTP, sehingga dapat menghasilkan algoritma yang mangkus untuk kompleksitas waktu dan ruang.

5. Simpulan

Dalam penelitian ini menunjukan bahwa algoritma yang di rancang sangat baik dalam membangkitkan bilangan acak dengan tingkat korelasi yang baik dalam kriptografi. Pengujian keacakan yang dilakukan dengan menggunakan metode *Run Test, Mono Bit*, dan *Block Bit* dengan program yang dibuat dalam *Microsoft Excel* selalu menghasilkan *p-value* kurang dari 0.01, dengan demikian setiap kunci yang diuji menghasilkan keluaran acak yang baik. Perbandingan pengujian dengan penelitian sebelumnya juga menghasilkan keluaran acak yang sangat baik. Pengujian enkripsi dengan kunci yang berbeda dengan tiga plainteks yang berbeda menghasilkan korelasi antar plainteks dan cipherteks yang baik dengan tingkat korelasi sangat rendah. Dengan demikian dapat disimpulkan bahwa pembangkit kunci LFSR dengan tiga blok bit fungsi XOR dapat menghasilkan nilai korelasi kunci yang baik dan keluaran acak yang aman digunakan dalam sebuah kriptografi.

Pada tiga penelitian sebelumnya yang dimana juga melakukan penelitian dengan melakukan perancangan algoritma yang menggunakan jumlah fungsi XOR yang lebih banyak dari penelitian ini. Terdapat penelitian yang merancang dengan menggunakan empat fungsi XOR, lima fungsi XOR dan tujuh fungsi XOR. Tiga penelitian terdahulu juga menyatakan hasil dimana rancangan algoritma yang digunakan dapat menghasilkan korelasi kunci dan keluaran acak yang sangat aman untuk digunakan dalam kriptografi. Sedangkan pada penelitian kali ini dirancang dengan menggunakan tiga fungsi XOR yang dimana sudah diuji bahwa algoritma yang dirancang dengan tiga fungsi XOR juga digunakan dengan sangat baik dalam membangkitkan bilangan acak dengan tingkat korelasi yang baik dalam kriptografi.

Pustaka

[1] Herman, A. J., *"Desain Pembangkit Kunci LFSR dengan Skema A5/1 Menggunakan 7 Blok Bit Fungsi XOR"*,(Doctoral dissertation), 2022.

[2] Nafurbenan, R. Maria, *"Perancangan Enam Bagan Fungsi XOR sebagai Umpan Balik sebagai Pembangkit Bilangan Acak LFSR dengan Skema A5/1"*.Skripsi S-1 Teknik, 2022

[3] Manullang, R. V., *"Desain Lima Fungsi Umpan Balik LFSR dengan Skema A5/1 sebagai Pembangkit Kunci Kriptografi Stream Cipher"*, (Doctoral dissertation), 2022.

[4] Nahading, T. S., & Wowor, A. D.., *"Desain Pembangkit Kunci LFSR dengan Skema A5/1 Menggunakan empat Blok Bit Fungsi XOR"*., Kesatria: Jurnal Penerapan Sistem Informasi (Komputer dan Manajemen), 4(2), 409-419, 2023.

[5] Hutagalung, W. H., *"Implementasi Vigenere Chiper Dengan Random Key Metode Linear Feedback Shift Register (LFSR) Pada Teks"*.,Informasi dan Teknologi Ilmiah (INTI), 4(3), 2017.

[6] Robby, F. C. *"Pengujian Statistik pada Pembangkit Kunci LFSR dalam Skema Algoritma A5/1 "*. (Doctoral dissertation), 2021.

[7] Kurniawan, D., & Priyatna, B., *"engamanan data berbasis mobile android dengan penggabungan linear feedback shift register (lfsr) dan modifikasi matriks kunci algoritma kriptografi playfair cipher"*., Telematika MKOM, 10(1), 42-46, 2018.

[8] O. D. Jensen, "A5 Encryption In GSM,", *n*o. June, pp. 3–8, 2017.

[9] Wowor, Alz Danny,, *"Regenerasi Fungsi Polinomial Dalam Rancangan Algoritma Berbasis CSPRNG Chaos Sebagai Pembangkit Kunci pada Kriptografi Block Cipher"*, Limits Journal, 14 : 4, 2017.

[10] A. Rukhin, J. Soto, and J. Nechvatal,, "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications,"no. April.

[11] Chapra, S.C., & Canale, R.P., *"Numerical Methods for Engineers"*, Sixth Edition, New York: McGraw-Hill.

.