Penerapan Serangan Linier pada Enkripsi Lengkap ALgoritme SIT-64

Aghisa Laelatu Zuhriyah[1], Yeni Farida, S.Stat., M.Si.[2]
Jurusan Teknik Persandian
Sekolah Tinggi Sandi Negara
Bogor, Indonesia
aghisa.laelatu@bssn.go.id[1], yeni.farida@stsn-nci.ac.id[2]

***Abstract*— Algoritme SIT-64 adalah algoritme yang diusulkan oleh Usman et al. pada jurnal *International Journal of Advanced Computer Science and Application* Vol. 8 No. 1 tahun 2017 yang digunakan untuk mengamankan data pada teknologi IoT. Algoritme SIT-64 diklaim tahan terhadap serangan linier, namun pada *paper* yang dipublikasikan tidak menyertakan hasil aproksimasi yang diperoleh. Pada penelitian ini, dilakukan penerapan serangan linier pada enkripsi lengkap algoritme SIT-64 untuk membuktikan ketahanan algoritme ini terhadap serangan linier. Penelitian ini berfokus pada proses pencarian aproksimasi linier yang dapat digunakan untuk proses *recovery* kunci. Pencarian dilakukan dengan cara mengonstruksi aproksimasi linier berdasarkan struktur dan sifat algoritme yang mempertimbangkan bias, probabilitas serta korelasi. Hasil yang diperoleh yaitu ditemukan 6 karakteristik *input/output masking* dengan 4 pola serangan. Berdasarkan hasil tersebut dapat diperoleh 12 aproksimasi linier pada 2-*round* dengan korelasi $\mathbf{C} = 2^{-3}$ serta dua aproksimasi linier 4-round yang dapat digunakan untuk proses *recovery* tujuh bit subkunci *round* kelima. Dua aproksimasi linier 4-*round* tersebut memiliki bias, probabilitas dan korelasi berturut-turut $|\boldsymbol{\varepsilon}| = 2^{-10}, \mathbf{Pr} = 0,4990234$ dan $\mathbf{C} = 2^{-9}$dengan kompleksitas data teoritis yang dibutuhkan untuk proses *recovery* kunci sebesar 2^{20}.**

Keywords*— Serangan linier, Algoritme SIT-64, Enkripsi data, IoT.

I. PENDAHULUAN

Enkripsi data merupakan salah satu mekanisme pengamanan terhadap serangan dalam proses transmisi data melalui internet yang memanfaatkan algoritme kriptografi [1]. Teknologi yang menggunakan transmisi data tersebut adalah teknologi *Internet of Things* (IoT). Algoritme kriptografi yang dimanfaatkan untuk melakukan enkripsi data pada teknologi IoT adalah algoritme berbasis *lightweight block cipher*. Dipilihnya algoritme berbasis *lightweight block cipher* karena ringan dalam implementasi [9].

SIT-64 merupakan algoritme kriptografi berbasis *lightweight block cipher* yang dirancang khusus untuk enkripsi data pada teknologi IoT [10]. Algoritme SIT-64 memiliki struktur *Generalized Feistel Network* (GFN). Ukuran blok teks terang/teks sandi Algoritme SIT-64 yaitu 64 bit yang terbagi menjadi empat blok berukuran 16 bit dan ukuran kunci 64 bit. Jumlah *round* untuk enkripsi lengkap adalah 5-*round* dan memerlukan lima buah subkunci berbeda untuk setiap *round*. Pada setiap *round* diperlukan dua subkunci sama. Pembuat algoritme menyatakan bahwa subkunci tersebut cukup kuat terhadap serangan kriptanalisis [10].

Komponen dalam suatu algoritme kriptografi terdiri dari komponen linier dan non-linier. Komponen non-linier pada algoritme SIT-64 adalah fungsi F. Struktur fungsi F pada algoritme SIT-64 terinspirasi dari *S-box* algoritme Khazad pada (Barreto, P & Rijmen, V., 2000) yang tersusun atas permutasi bit dan dua buah S-box semi acak berukuran 4 x 4.

Fungsi F yang merupakan komponen non-linier algoritme berstruktur Feistel menjadi objek utama dalam serangan linier [5]. Serangan linier merupakan suatu metode kriptanalisis menggunakan asumsi *known-plaintext attack* untuk mencari keterkaitan antara teks terang, teks sandi dan kunci [5]. Fokus utama dari serangan tersebut adalah mencari persamaan linier dari bit-bit teks terang, teks sandi dan kunci yang melewati komponen non-linier suatu algoritme dengan probabilitas $p \neq 12$, kemudian memanfaatkan informasi yang diperoleh untuk melakukan *recovery* kunci [6]. Oleh karena itu dalam serangan linier terdiri dari dua tahapan, yaitu pencarian aproksimasi linier dan *recovery* kunci [12]. Usman et al (2017) menyatakan bahwa algoritme SIT-64 tahan terhadap serangan linier untuk enkripsi lengkap dan pada aproksimasi linier 2-*round* terdapat korelasi sangat besar antara *input* dan *output*. Namun dalam *paper* yang dipublikasikan di jurnal *International Journal of Advanced Computer Science and Application* Vol. 8 No. 1 tahun 2017 tidak menyertakan hasil aproksimasi yang diperoleh.

Berdasarkan hal tersebut, pada penelitian ini akan dilakukan penerapan serangan linier pada algoritme SIT-64 dengan mencari aproksimasi linier 2-*round* dan aproksimasi linier 4-*round*. Fokus penelitian ini terdapat pada tahap pencarian aproksimasi linier serta analisis untuk mencari aproksimasi efektif untuk proses *recovery* kunci. Pencarian aproksimasi linier menggunakan pendekatan terhadap fungsi F dan struktur umum algoritme SIT-64. Aproksimasi linier tersebut digunakan untuk mengetahui ketahanan algoritme SIT-64 terhadap serangan linier.

Pembahasan hasil penelitian disusun menjadi lima bagian. Pada bagian pertama dibahas latar belakang penelitian. Bagian kedua membahas tentang metode penelitian. Bagian inti dari penelitian ini terletak pada bagian ketiga yaitu bagian hasil dan pembahasan yang berisi proses pencarian aproksimasi linier 4-*round* algroitme SIT-64 serta analisis yang dilakukan untuk memperoleh aproksimasi efektif yang dapat digunakan pada

proses *recovery* kunci. Kesimpulan dan saran untuk penelitian selanjutnya disebutkan pada bagian empat serta bagian lima berisi Referensi yang menjadi rujukan penelitian.

II. METODE

Metode penelitian yang digunakan pada penelitian ini adalah metode studi literatur dan metode eksperimen. Metode studi literatur digunakan untuk mempelajari konsep-konsep yang diperlukan pada penelitian ini antara lain GFN, algoritme SIT-64, dan serangan linier.

Metode eksperimen dilakukan dalam proses pencarian aproksimasi linier. Proses pencarian aproksimasi linier algoritme SIT-64 dilakukan dengan mencari semua kemungkinan aproksimasi linier yang terjadi berdasarkan pembatasan masalah dengan memperhatikan bias, korelasi dan probabilitas. Metode eksperimen juga dilakukan pada tahapan setelah mencari aproksimasi linier yaitu menganalisis hasil aproksimasi yang diperoleh untuk mengetahui ketahanan algoritme SIT-64 terhadap serangan linier. Berikut langkah-langkah yang dilakukan:

a. Menganalisis komponen non-linier pada algoritme SIT-64.

b. Mencari LAT dari komponen non-linier algoritme SIT-64.

c. Mengonstruksi aproksimasi linier 4-*round* algoritme SIT-64 dengan rincian tahapan sebagai berikut:

- Mengonstruksi pola serangan linier pada struktur algoritme SIT-64;
- Menentukan kemungkinan *input* dan *output masking* dari *S-box* dengan LAT ±4;
- Mengonstruksi *trail* linier dalam komponen fungsi F, contoh *trail* adalah sebagai berikut:

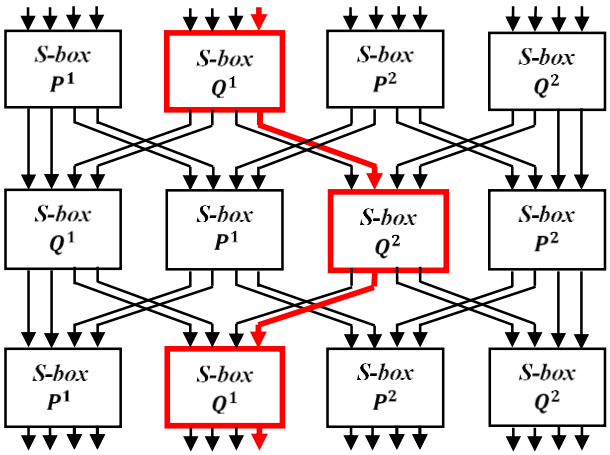

Fig. 1 Contoh *trail* linier dalam fungsi

Ketentuan yang digunakan untuk mengonstruksi *trail* linier tersebut terdapat pada Tabel 1.

TABEL 1 Ketentuan *input* dan *output masking S-box*

Layer ke -	**Ketentuan *input* dan *output masking s-box***				
	w_t *input masking S-box*	w_t *output masking S-box*	LAT *S-box*	Posisi bit *output masking S-box*	Jumlah *S-box* pada *layer* selanjutnya
1	Tidak dibatasi	$w_t \leq 2$	±4	$P_{1,2}; P_{3,4}; Q_{1,2}$ atau $Q_{3,4}$	1 buah
2	$w_t \leq 2$				
3		Tidak dibatasi		Tidak ditentukan	

- Menghitung probabilitas dan bias untuk *trail* linier fungsi F;
- Mengonstruksi aproksimasi linier 4-*round*;

  Proses ini merupakan kolaborasi dari tahap konstruksi pola serangan dan konstruksi *input* dan *output masking* fungsi F. Konstruksi aproksimasi linier 4-*round* diperoleh dari aproksimasi linier 3-*round* begitu pula aproksimasi linier 3-*round* diperoleh dari aproksimasi 2-*round* hingga seterusnya. Pada tahap ini, dilakukan pencarian aproksimasi linier 4-*round* untuk seluruh variasi *input* dan *output masking* fungsi F yang memenuhi ketentuan pada semua pola serangan linier.
- Menghitung probabilitas, bias dan korelasi setiap aproksimasi linier;

  Perhitungan bias sesuai dengan kaidah *pilling-up lemma* [5] dan korelasi sesuai definisi Knudsen & Robshaw (2011).
- Menganalisis aproksimasi linier 4-*round* algoritme SIT-64;

  Analisis dilakukan berdasarkan pengaruh *input* dan *output masking* fungsi F terhadap jumlah subkunci yang akan dilakukan *recovery* serta pengaruh pola serangan linier terhadap waktu simulasi *recovery* subkunci. Analisis yang dilakukan bertujuan untuk memperoleh aproksimasi yang efektif untuk digunakan pada proses *recovery* kunci.

III. HASIL DAN PEMBAHASAN

A. *Pencarian Aproksimasi Linier*

Proses pencarian aproksimasi linier terbagi menjadi 7 tahapan yang secara garis besar telah dijelaskan pada Bagian Metode. Aproksimasi linier dikonstruksi dari karakteristik *input*/*output* fungsi F terpilih yang diterapkan pada pola serangan terpilih. Berdasarkan analisis komponen yang dilakukan, diperoleh hasil bahwa komponen linier pada algoritme SIT-64 hanya pada layer permutasi dalam struktur fungsi F. Operasi XNOR bit serta layer konfusi dalam struktur fungsi F tergolong non-linier. Percobaan yang dilakukan pada operasi XNOR menunjukkan bahwa operasi XNOR merupakan

fungsi *affine* dari fungsi linier XOR atau *invers* dari operasi XOR bit. Hal tersebut dibuktikan secara matematis sebagai berikut:

Pembuktian bahwa komponen operasi XNOR bit subkunci *round* merupakan komponen linier atau non-linier dapat dijelaskan sebagai berikut: misal dua buah blok teks terang berukuran 16 bit yaitu Px_i dan $P'x_i$ serta subkunci $K_{p_z}^j$ berukuran 16 bit sebagai berikut:

- $Px_z = x_1x_2x_3x_4x_5x_6x_7x_8x_9x_{10}x_{11}x_{12}x_{13}x_{14}x_{15}x_{16}$
- $Px'_z = x'_1x'_2x'_3x'_4x'_5x'_6x'_7x'_8x'_9x'_{10}x'_{11}x'_{12}x'_{13}x'_{14}x'_{15}x'_{16}$
- $K_{p_z}^j = k_{p_1}^j k_{p_2}^j k_{p_3}^j k_{p_4}^j k_{p_5}^j k_{p_6}^j k_{p_7}^j k_{p_8}^j k_{p_9}^j k_{p_{10}}^j k_{p_{11}}^j k_{p_{12}}^j k_{p_{13}}^j k_{p_{14}}^j k_{p_{15}}^j k_{p_{16}}^j$

Jika $f(x_i, k_{p_z}^j)$ adalah fungsi operasi XNOR bit antara kunci $K_{p_z}^j$ dengan blok teks terang Px_z dan $f(x'_i, k_{p_z}^j)$ merupakan fungsi operasi XNOR bit antara kunci $K_{p_z}^j$ dengan blok teks terang Px'_z, maka akan dibuktikan bahwa $f((x_z \oplus k_{p_z}^j), (x'_z \oplus k_{p_z}^j)) = f(x_z, k_{p_z}^j) \oplus f(x'_z, k_{p_z}^j)$ (1)

$$
\begin{aligned}
f[(x_z \oplus k_{p_z}^j), (x'_z \oplus k_{p_z}^j)] &= f[(x_1 \oplus k_{p_1}^j, \dots x_{16} \oplus k_{p_{61}}^j), (x'_1 \oplus k_{p_1}^j, \dots x'_{16} \oplus k_{p_{16}}^j)] \\
&= (x_1 \oplus k_{p_1}^j, \dots x_{16} \oplus k_{p_{16}}^j) \odot (x'_1 \oplus k_{p_1}^j, \dots x'_{16} \oplus k_{p_{16}}^j) \\
&= \left((x_1 \oplus k_{p_1}^j) \odot (x'_1 \oplus k_{p_1}^j)\right) \dots ((x_{16} \oplus k_{p_{16}}^j) \odot (x'_{16} \oplus k_{p_{16}}^j)) \\
&= (x_1 \oplus k_{p_1}^j \oplus x'_1 \oplus k_{p_1}^j \oplus 1), \dots (x_{16} \oplus k_{p_{16}}^j \oplus x'_{16} \oplus k_{p_{16}}^j \oplus 1) \\
&= (x_z \oplus x'_z \oplus k_{p_z}^j \oplus k_{p_z}^j \oplus 1) \\
&= (x_z \oplus x'_z \oplus 1) \blacksquare
\end{aligned}
$$

$$
\begin{aligned}
f(x_z, k_{p_z}^j) \oplus f(x'_z, k_{p_z}^j) &= (x_z \odot k_{p_z}^j) \oplus (x'_z \odot k_{p_z}^j) \\
&= (x_1 \odot k_{p_1}^j, \dots x_{16} \odot k_{p_{16}}^j) \oplus (x'_1 \odot k_{p_1}^j, \dots x'_{16} \odot k_{p_{16}}^j) \\
&= (x_1 \odot k_{p_1}^j) \oplus (x'_1 \odot k_{p_1}^j), \dots (x_{16} \odot k_{p_{16}}^j) \oplus (x'_{16} \odot k_{p_{16}}^j) \\
&= (x_1 \oplus k_{p_1}^j \oplus 1 \oplus x'_1 \oplus k_{p_1}^j \oplus 1), \dots \\
&\quad (x_{16} \oplus k_{p_{16}}^j \oplus 1 \oplus x'_{16} \oplus k_{p_{16}}^j \oplus 1) \\
&= (x_z \oplus x'_z \oplus k_{p_z}^j \oplus k_{p_z}^j \oplus 1 \oplus 1) \\
&= (x_z \oplus x'_z) \blacksquare
\end{aligned}
$$

Pembuktian terhadap komponen *layer* konfusi fungsi F yaitu *S-box* P dan Q dilakukan secara kontradiksi dengan mengambil sebarang *input S-box* kemudian membuktikan bahwa *S-box* tidak bersifat linier. Berdasarkan bukti tersebut, selanjutnya dilakukan konstruksi LAT dari operasi XNOR bit dan *layer* konfusi fungsi F yaitu *S-box* P dan Q. Selanjutnya dilakukan analisis terhadap LAT yang dikonstruksi dan diperoleh hasil sebagai berikut:

a. Pada komponen XNOR bit terdapat pengaruh dari *hamming weight* (w_t) *output* operasi XNOR bit yaitu apabila w_t berjumlah genap, maka LAT yang diperoleh akan maksimal dengan probabilitas 1 sedangkan apabila sebaliknya maka LAT yang diperoleh akan maksimal namun dengan probabilitas 0. Hal ini berlaku untuk 2 *input* (α) dan *output* (β) yaitu $\alpha = \beta_{16} || \beta_{16}$ dan β_{16}.
b. Pada komponen *S-box* P dan Q dapat diidentifikasi bahwa untuk setiap *input masking* tak nol dari masing-masing *S-box* akan menghasilkan dua buah *output masking* dengan LAT ± 4. Tabel 2 menunjukkan hasil identifikasi terhadap *S-box* P dan Q.

TABEL 2 Daftar *input* dan *output masking S-box* (dalam heksadesimal) dengan LAT ± 4

α		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	***a***	***b***	***c***	***d***	***e***	***f***
β	**P**	1	5	8	*b*	2	6	7	3	7	8	4	6	1	3	4
		d	*d*	*e*	*f*	5	*c*	9	*a*	9	*e*	*f*	*c*	2	*a*	*b*
	Q	1	2	8	4	*b*	6	1	3	4	3	5	6	2	5	8
		7	*d*	*a*	9	*e*	*c*	9	*f*	7	*f*	*e*	*c*	*d*	*b*	*a*

Hasil konstruksi pola serangan berdasarkan struktur algoritme SIT-64 menunjukkan bahwa terdapat empat pola serangan yang digunakan untuk mengonstruksi aproksimasi linier, pola tersebut disebut sebagai pola 1, pola 2, pola 3 dan pola 4 dengan ketentuan sebagai berikut:

TABEL 3 Pola serangan yang digunakan

No	**Pola Serangan**	**Posisi blok teks terang aktif**	**Karakteristik *output masking round* ke-4**
1	Pola 1	$\alpha 0 \beta 0$	$\alpha\alpha\alpha 0_{16}$
2	Pola 2	$0\alpha 0\beta$	$0\alpha\alpha\alpha_{16}$
3	Pola 2	$\alpha\beta 0\gamma$	$\alpha 0\alpha 0_{16}$
4	Pola 4	$\alpha 0\beta\gamma$	$0\alpha 0\alpha_{16}$

Setelah mengonstruksi pola serangan, langkah selanjutnya adalah melakukan analisis pendekatan linier terhadap komponen fungsi F dengan cara mengonstruksi LAT *S-box* P dan Q, kemudian mengonstruksi pola *S-box* aktif yaitu pola 1-1-1, dilanjutkan dengan mengonstruksi *input* dan *output masking S-box* masing-masing *layer* hingga ditemukan *input/output masking* fungsi F.

Dari 120 *input masking* tak nol di *layer* konfusi pertama, diperoleh 40 *output masking* pada *layer* konfusi ketiga. Selanjutnya dilakukan konstruksi *trail* linier dalam fungsi F dan ditemukan 78 *trail* linier *input/output masking* fungsi F. Dari 78 *trail* tersebut dapat diidentifikasi enam kondisi karakteristik *input/output masking* fungsi F, antara lain:

1. Kondisi 1 : $\alpha \rightarrow \beta$ dan $\beta \rightarrow \alpha$;
2. Kondisi 2 : $\alpha \rightarrow \beta$ dan $\beta \rightarrow \gamma$;
3. Kondisi 3 : $\alpha \rightarrow \beta$ dan $\beta \rightarrow \beta$;
4. Kondisi 4 : $\alpha \rightarrow \alpha$ dan $\alpha \rightarrow \beta$;
5. Kondisi 5 : $\beta \rightarrow \delta$, $\delta \rightarrow \gamma$ dan $\gamma \rightarrow \alpha$;
6. Kondisi 6 : $\alpha \rightarrow \alpha$.

Dilihat dari sudut pandang penyerang, kondisi 6 adalah kondisi yang menguntungkan bagi penyerang karena penyerang hanya membutuhkan satu jenis *input/output masking* yaitu α dengan karakteristik pemetaan fungsi F : $\alpha \rightarrow \alpha$. Penyerang tidak perlu mencari *output masking* yang berbeda dengan *input masking*. Hal ini membuat penyerang lebih mudah dan efektif untuk mencari *input/output masking* yang dibutuhkan. Jumlah

karakteristik *input/output masking* yang memenuhi kondisi 6 tersebut ditunjukkan pada Tabel 4.

TABEL 4 Daftar karakteristik *input/output masking* fungsi F

No	*Input masking* (α)	Pola *S-box* aktif	*Output masking* (α)
1.	$b000$	$(P^1_{1,3,4}, P^1_2) - (Q^1_2, Q^1_2) - (P^1_2, P^1_{1,3,4})$	$b000$
2.	0001	$(Q^2_4, Q^2_4) - (P^2_4, P^2_4) - (Q^2_4, Q^2_4)$	0001
3.	0007	$(Q^2_{2,3,4}, Q^2_4) - (P^2_4, P^2_4) - (Q^2_4, Q^2_{2,3,4})$	0007
4.	6000	$(P^1_{2,3}, P^1_{1,2}) - (Q^1_{1,2}, Q^1_{1,2}) - (P^1_{1,2}, P^1_{2,3})$	6000
5.	$c000$	$(P^1_{1,2}, P^1_{1,2}) - (Q^1_{1,2}, Q^1_{1,2}) - (P^1_{1,2}, P^1_{1,2})$	$c000$
6.	$f000$	$(P^1_{1,2,3,4}, P^1_2) - (Q^1_2, Q^1_2) - (P^1_2, P^1_{1,2,3,4})$	$f000$

Selanjutnya dilakukan konstruksi aproksimasi linier 4-*round* menggunakan seluruh karakteristik *input/output masking* fungsi F yang memenuhi. Contoh konstruksi aproksimasi linier 4-*round* menggunakan karakteristik *input/output masking* fungsi F 0001_{16} dapat dilihat pada Gambar 2 dan hasil dari konstruksi aproksimasi linier 4-*round* secara keseluruhan dapat dilihat pada Bab II.

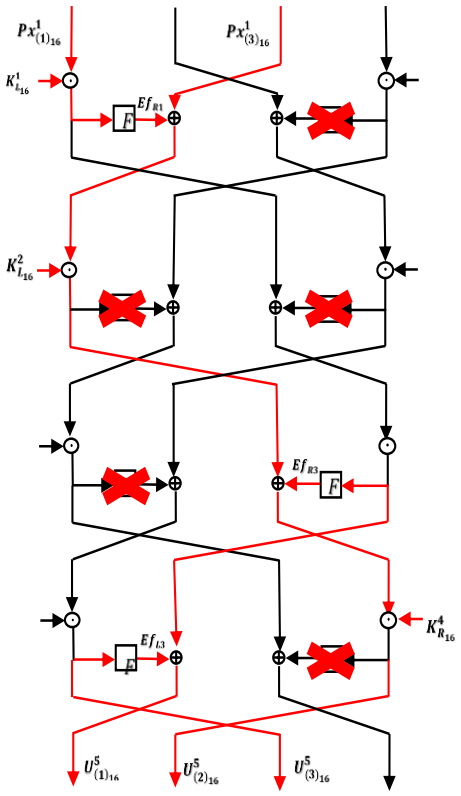

Fig. 2 Contoh *trail* aproksimasi linier 4-round

B. Hasil Pencarian Aproksimasi Linier 4-round

Bagian ini berisi tentang hasil karakteristik *output masking* 4-*round*, bias, probabilitas dan korelasi yang diperoleh dari konstruksi aproksimasi linier untuk semua karakteristik *input masking*. Tabel 5 adalah tabel karakteristik *output masking* 4-*round* untuk seluruh pola serangan dan seluruh variasi *input/output masking* fungsi F.

TABEL 5 Karakteristik *output masking* 4-*round*

No	Karakteristik *input/output masking* fungsi F	Karakteristik *output masking round* ke-4	
		Pola 1	**Pola 2**
1	$b000$	$b000b000b0000000_{16}$	$0000b000b000b000_{16}$
2	0001	0001000100010000_{16}	0000000100010001_{16}
3	0007	0007000700070000_{16}	0000000700070007_{16}
4	6000	6000600060000000_{16}	0000600060006000_{16}
5	$c000$	$c000c000c0000000_{16}$	$0000c000c000c000_{16}$
6	$f000$	$f000f000f0000000_{16}$	$0000f000f000f000_{16}$
		Pola 3	**Pola 4**
1	$b000$	$b0000000b0000000_{16}$	$0000b0000000b000_{16}$
2	0001	0001000000010000_{16}	0000000100000001_{16}
3	0007	0007000000070000_{16}	0000000700000007_{16}
4	6000	6000000060000000_{16}	0000600000006000_{16}
5	$c000$	$c0000000c0000000_{16}$	$0000c0000000c000_{16}$
6	$f000$	$f0000000f0000000_{16}$	$0000f0000000f000_{16}$

Berdasarkan Tabel di atas terlihat perbedaan karakteristik *output masking* untuk setiap pola serangan dikarenakan perbedaan variasi *input/output masking* fungsi F. Hal ini juga mengakibatkan adanya perbedaan posisi bit subkunci yang telibat dalam aproksimasi linier. Namun, bias, probabilitas dan korelasi akan bernilai sama dalam satu pola serangan meskipun berbeda variasi *input/output masking* fungsi F. Oleh karena itu nilai-nilai tersebut dibedakan berdasarkan pola serangan seperti pada Tabel 6.

TABEL 6 Bias, probabilitas dan korelasi setiap pola serangan

No	*Round*	**Pola 1 dan 2**		
		Bias	**Probabilitas**	**Korelasi**
1.	1	$\frac{1}{2^4}$	0,484375	$\frac{1}{2^3}$
2.	2	$\frac{1}{2^4}$	0,484375	$\frac{1}{2^3}$
3.	3	$\frac{1}{2^7}$	0,4921875	$\frac{1}{2^6}$
4.	4	$\frac{1}{2^{10}}$	0,4990234	$\frac{1}{2^9}$
		Pola 3 dan 4		
1.	1	$\frac{1}{2^4}$	0,484375	$\frac{1}{2^3}$
2.	2	$\frac{1}{2^7}$	0,4921875	$\frac{1}{2^6}$
3.	3	$\frac{1}{2^7}$	0,4921875	$\frac{1}{2^6}$
4.	4	$\frac{1}{2^{10}}$	0,4990234	$\frac{1}{2^9}$

Berdasarkan hasil yang diperoleh, terdapat dua nilai yang berbeda untuk bias, probabilitas dan korelasi diantara empat pola sehingga dilakukan pengelompokan pola berdasarkan bias, probabilitas serta korelasi yang sama. Terlihat bahwa pola 1 dan pola 2 memiliki nilai yang sama untuk setiap *round* begitu pula pola 3 dan 4. Perbedaan antara pola 1 dan 2 dengan pola 3 dan 4 terletak pada *round* kedua yaitu pada pola 1 dan 2 dihasilkan nilai-nilai yang lebih besar dibandingkan pola 3 dan 4. Hal itu disebabkan karena pada *round* kedua, pola 1 dan 2 tidak mengaktifkan fungsi F di kedua sisi sehingga bias total yang dihasilkan akan lebih besar dibandingkan pola 3 dan 4 yang mengaktifkan fungsi F di salah satu sisi. Hasil ini membuktikan bahwa bias total dipengaruhi oleh jumlah fungsi F yang aktif.

C. *Analisis Efektifitas Aproksimasi Linier 4-round*

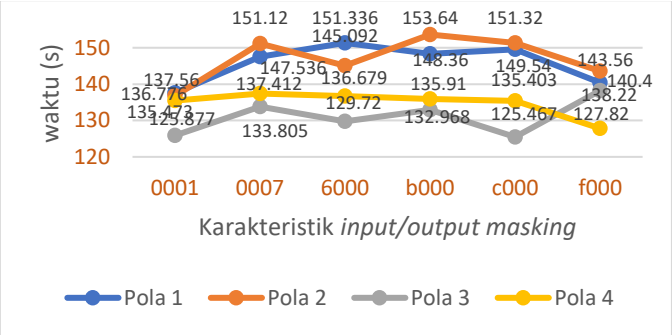

Fig. 3 Grafik waktu simulasi semua pola untuk sampel 4

Berdasarkan Gambar 3, pola serangan paling efektif adalah pola 3 dengan rata-rata waktu simulasi sebesar 131,01 detik. Hampir semua karakteristik yang digunakan pada pola 3 memiliki waktu simulasi tercepat dibandingkan pola lain. Efektifitas dari aproksimasi linier 4-*round* tidak hanya berdasarkan waktu simulasi saja, namun juga dilihat dari jumlah subkunci K_p^5 yang dapat ditebak. Semakin singkat waktu yang dibutuhkan dan semakin banyak bit subkunci yang di-*recovery* maka dikatakan semakin efektif aproksimasi linier tersebut.

D. *Analisis Resistensi Algoritme SIT-64*

Berdasarkan analisis yang telah dilakukan pada bagian A sampai dengan C, maka dapat dikatakan bahwa secara teoritis, untuk memperoleh kunci yang benar dibutuhkan kompleksitas data (N) $= 2^{20}$, kompleksitas data tersebut jauh lebih kecil dibandingkan dengan kompleksitas data minimal algoritme SIT-64 untuk dikatakan *computationally secure* yaitu sebesar 2^{64}. Maka dapat dikatakan bahwa secara teoritis algoritme SIT-64 rentan terhadap serangan linier.

IV. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian, dapat disimpulkan sebagai berikut:

a. Ditemukan 12 aproksimasi linier 2-round yang memiliki korelasi maksimal yaitu $C = \frac{1}{2^3}$ dengan karakteristik input masking $\alpha 0 \alpha 0$ untuk pola serangan linier 1 dan karakteristik input masking $0\alpha 0\alpha$ untuk pola serangan linier 2;
b. Ditemukan 2 aproksimasi linier 4-*round* dengan bias maksimal $|\varepsilon| = \frac{1}{2^{10}}$, Pr $= 0{,}4990234$ dan korelasi $C = \frac{1}{2^9}$ yang digunakan untuk proses *recovery* subkunci dengan kompleksitas data yang dibutuhkan adalah $N \approx 2^{20}$.

Saran untuk pengembangan selanjutnya adalah sebagai berikut:

a. Melakukan pencarian aproksimasi linier lain menggunakan pola *S-box* fungsi F selain pola 1-1-1 serta mencari *linear hulls* yang dapat digunakan untuk mengonstruksi aproksimasi linier;
b. Mencoba seluruh kemungkinan *input* dan *output* fungsi F untuk memperoleh aproksimasi linier lain yang dapat digunakan untuk me-*recovery* lebih banyak bit subkunci.

UCAPAN TERIMAKASIH

Pada bagian ini penulis mengucapkan terimakasih kepada pihak civitas akademika Sekolah Tinggi Sandi Negara dan kepada pihak-pihak yang membantu penulis menyelesaikan penelitian.

REFERENSI

[1] W. Stallings. Cryptography and Network Security Principles and Practice. Fifth Ed. New York: Pearson Education. 2011.

[2] Knudsen, L. R., & Robshaw, M. J. The Block Cipher Companion. New York: Springer. 2011.

[3] O'Connor, L. Properties of Linear Approximation Tables. Brisbane: Springer. 1995.

[4] Heys, H. A Tutorial on Linear dan Differential Cryptanalysis. Electrical and Computer Engineering Faculty of Engineering and Applied Science Memorial University of Newfoundland. Canada. 2001.

[5] Matsui, M. "Linear Cryptanalysis Method for DES Cipher". EUROCRYPT'93, 1993, pp. 386-397. Berlin: Springer-Verlag.

[6] Robshaw, M., & Kaliski, B. "Linear Cryptanalysis Using Multiple Approximations". Advances in Cryptology - CRYPTO'94, 1994, pp. 1-11. Springer-Verlag.

[7] Schneier, B., & Kelsey, J." Unbalanced Feistel Networks and Block Cipher Design". Fast Software Encryption, 1996, pp. 121-144. Springer-Verlag.

[8] Biham, E. "On Matsui's Linear Cryptanalysis". Advances in Cryptology — EUROCRYPT'94. EUROCRYPT 1994. Lecture Notes in Computer Science, vol 950. 1995. Berlin: Springer.

[9] Mohd, B., & Hayajneh, T. "Lightweight Block Ciphers for IoT: Energy Optimization and Survivability Techniques". IEEE, 2018, pp. 2169-3536. doi:10.1109/ACCESS.2018.2848586.

[10] Usman *et* al., "SIT : A Lightweight Encryption Algorithm for Secure Internet of Things". International Journal of Advanced Computer Science and Applications. 2017. doi:10.14569/IJACSA.2017.080151.

[11] Hoang, V. T., & Rogaway, P. "On Generalized Feistel Network". International Association for Cryptologic Research, 2010, pp. 613–630. doi:10.1007/978-3-642-14623-7_33.

[12] Mirza, F. (1998). *Block Ciphers and Cryptanalysis.* www.citeseer.nj.nec.com/266836.html.