Volume 14 Nomor 1 – Maret 2020

Info KriptO

Jurnal Ilmiah Keamanan Siber dan Kriptologi

ISSN 1978 - 7723

ISSN 1978-7723
Maret 2020
Halaman 1 - 60
Nomor 1
Volume 14
INFO KRIPTO

Pusat Penelitian dan Pengabdian Masyarakat
Politeknik Siber dan Sandi Negara (Poltek SSN)

ISSN 1978-7723

Info KriptO

Jurnal Ilmiah Keamanan Siber dan Kriptologi

Jurnal Info Kripto dipublikasikan oleh Pusat Penelitian dan Pengabdian Masyarakat Politeknik Siber dan Sandi Negara. Jurnal ini diterbitkan dari hasil penelitian dari berbagai bidang yang terkait dengan Keamanan Siber, Keamanan Informasi dan Kriptologi.

SEKRETARIAT
Jurnal Info Kripto – Pusat Penelitian dan Pengabdian Masyarakat
Politeknik Siber dan Sandi Negara (Poltek SSN)
Telp (0251) 8541754, Fax (0251) 8541720
Email: infokripto@poltekssn.ac.id

PENGANTAR REDAKSI

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa, karena atas limpahan rahmat dan karunia-Nya kami dapat menyelesaikan jurnal Info Kripto Volume 14 Nomor 1 Tahun 2020. Penyelesaian Jurnal Info Kripto ini merupakan hasil kerja sama berbagai pihak. Oleh karena itu, Kami selaku tim redaksi, mengucapkan terima kasih kepada penulis, editor dan semua pihak yang tidak dapat kami sebutkan satu persatu, yang telah membantu penerbitan jurnal Info Kripto ini.

Jurnal Info Kripto merupakan salah satu sarana publikasi hasil penelitian civitas akademik, khususnya yang berada di lingkungan Poltek SSN. Jurnal Info Kripto yang diterbitkan sebanyak tiga kali dalam setahun memuat hasil penelitian yang bertemakan kriptologi, keamanan informasi dan keamanan siber. Hadirnya Jurnal Info Kripto ini diharapkan dapat meningkatkan budaya penelitian yang kondusif dan komprehensif di lingkungan Poltek SSN.

Kami menyadari bahwa jurnal Info Kripto ini masih terdapat banyak kekurangan. Oleh karena itu, kami selalu terbuka terhadap kritik dan saran yang bersifat membangun untuk penerbitan jurnal Info Kripto yang lebih baik lagi pada edisi selanjutnya.

Bogor, Maret 2020

Redaktur

DAFTAR ISI

Kajian Matematis Basis Gröbner sebagai Penyelesaian Sistem Persamaan Polinomial Multivariat pada Serangan Aljabar

Ahmad Balya Izzuddin[1)] dan Sri Rosdiana[2)]
(1) *Politeknik Siber dan Sandi Negara / ahmad.balya@student.poltekssn.ac.id*
(2) *Politeknik Siber dan Sandi Negara / sri.rosdiana@poltekssn.ac.id*

Abstrak

Serangan aljabar merupakan proses memecahkan suatu teks sandi dengan cara menyelesaikan sistem persamaan polinomial. Salah satu metode penyelesaian sistem persamaan polinomial adalah dengan metode basis Gröbner. Konsep dasar dari basis Gröbner adalah reduksi polinomial untuk menghitung dan mendefinisikan bentuk normal yang sesuai dari suatu polinomial. Pada penelitian ini dilakukan kajian matematis terhadap penggunaan basis Gröbner sebagai penyelesaian sistem persamaan polinomial multivariat. Selain itu, juga dilakukan penyelesaian terhadap sistem persamaan polinomial yang telah diperoleh dari serangan aljabar. Sistem persamaan polinomial yang digunakan merupakan hasil serangan terhadap algoritme Geffe generator. Pada penelitian ini, sistem persamaan polinomial berhasil direduksi dan seluruh nilai seed *berhasil diperoleh. Dengan demikian, dapat disimpulkan bahwa Basis Gröbner dapat diterapkan dalam menyelesaikan sistem persamaan polinomial multivariat yang diperoleh dari serangan aljabar.*

Kata Kunci: *Basis Gröbner (1), Kajian Matematis (2), Serangan Aljabar (3), Polinomial Multivariat (4).*

1. PENDAHULUAN

Kriptanalisis merupakan studi mengenai teknik matematika yang bertujuan memecahkan suatu sistem kriptografi. Kriptanalisis pada algoritme simetrik dibagi menjadi tiga, yaitu *ciphertext-only attack*, *known-plaintext attack*, dan *chosen plaintext attack* [1]. Kriptanalisis juga dapat digunakan untuk mencari kelemahan pada sebuah sistem kriptografi [2].

Salah satu metode kriptanalisis adalah *algebraic cryptanalysis* yaitu proses memecahkan suatu teks sandi dengan cara menyelesaikan sistem persamaan polinomial [3]. Terdapat beberapa faktor yang memengaruhi proses serangan aljabar, antara lain jumlah variabel, jumlah polinomial, jumlah derajat polinomial, kekuatan penyelesaian persamaan, kecepatan dan jumlah memori komputer yang digunakan [4]. Salah satu tahapan dalam serangan aljabar adalah penyelesaian sistem persamaan polinomial. Untuk dapat menyelesaikan sistem persamaan ini terdapat beberapa metode yang dapat digunakan, antara lain basis Gröbner, *Linearization*, *Extended Linearization* (XL) *Algorithm*, dan ElimLin [3].

Penyelesaian sistem persamaan polinomial dapat dilakukan dengan metode basis Gröbner. Basis Gröbner pertama kali diperkenalkan oleh Bruno Buchberger dalam penelitiannya pada tahun 1965 [5], yang kemudian diterjemahkan ke dalam Bahasa Inggris pada tahun 2006 [6]. Teori basis Gröbner berfokus pada konsep ideal yang dibangun oleh himpunan polinomial multivariat berhingga. Konsep dasar dari basis Gröbner adalah ide reduksi polinomial untuk menghitung dan mendefinisikan bentuk normal yang sesuai dari suatu polinomial [7].

Pada penelitian ini dilakukan kajian matematis terhadap penggunaan basis Gröbner sebagai penyelesaian sistem persamaan polinomial multivariat. Selain itu untuk menambah pemahaman mengenai penerapan basis Gröbner dalam serangan aljabar. Pada penelitian ini juga dilakukan penyelesaian terhadap sistem persamaan polinomial yang telah diperoleh menggunakan serangan aljabar. Sistem persamaan polinomial yang digunakan merupakan hasil serangan terhadap algoritme *Geffe generator*.

2. LANDASAN TEORI

2.1. Polinomial Multivariat

Polinomial multivariat merupakan polinomial dengan lebih dari satu variabel.

Definisi 2.1.1 [8]**:** Setiap polinomial f dengan n buah variabel dapat dituliskan sebagai hasil penjumlahan

$$f = \sum_{v \in \mathbb{N}^n} a_v X^v$$

dengan $a_v \in R$ dan $v = (v_1, \dots, v_n) \in \mathbb{N}^n$. Himpunan seluruh polinomial dalam n buah variabel $X_1, \dots, X_n$ dinotasikan dengan $R[X_1, \dots, X_n]$.

Bentuk X^v dengan $v = (v_1, \dots, v_n) \in \mathbb{N}^n$, disebut sebagai monomial [9]. Sedangkan bentuk $rX^v \in R[\mathbb{N}^n]$ disebut *term*, dengan $r \in R \backslash \{0\}$ disebut sebagai koefisien [8].

Untuk dapat mengurutkan monomial atau *term* dalam polinomial multivariat, digunakan *term ordering* atau *monomial ordering* yang didefinisikan sebagai berikut [8]:

Definisi 2.1.2 *[8]***:** Suatu partial ordering $\leq$ pada $\mathbb{N}^n$ disebut term ordering jika untuk setiap $v, v_1, v_2 \in \mathbb{N}^n$ berlaku:

i. relasi $\leq$ merupakan *total ordering*;
ii. $0 \leq v$; dan

iii. $v_1 \leq v_2 \Rightarrow v_1 + v \leq v_2 + v$.

Contoh 2.1.3 *[8]*: Didefinisikan *lexicographic ordering* $\leq_{lex}$ pada $\mathbb{N}^n$ sebagai

$$(v_1, \dots, v_n) \leq_{\text{lex}} (w_1, \dots, w_n)$$

jika berlaku salah satu dari kondisi berikut:
$(v_1 < w_1)$ atau
$(v_1 = w_1)$ dan $(v_2 < w_2)$ atau
$(v_1 = w_1)$ dan $(v_2 = w_2)$ dan $(v_3 < w_3)$ atau
$\vdots$
$(v_1 = w_1)$ dan $(v_2 = w_2)$ dan ... dan $(v_n < w_n)$.
Ini tidak lain adalah urutan "alfabetis" pada tupel bilangan asli, contoh, $(1,2,3) \geq_{\text{lex}} (1,1,3)$ karena $2 > 1$, dan $(4,5,1) \leq_{\text{lex}} (4,5,3)$ karena $1 < 3$.

Definisi 2.1.4 *[8]*: Misalkan diberikan suatu polinomial multivariat

$$f = \sum_{v \in \mathbb{N}^n} a_v X^v$$

yang merupakan polinomial tak nol di $R[\mathbb{N}^n]$ dan $\leq$ merupakan *term ordering* pada $\mathbb{N}^n$. *Initial term* dari f berdasarkan $\leq$ didefinisikan sebagai

$$\text{in}_{\leq}(f) = a_w X^w$$

dengan $w = \max_{\leq}\{v \in \mathbb{N}^n | a_v \neq 0\}$.

Lema 2.1.5 (Dickson) *[8]*: Misalkan S himpunan bagian dari $\mathbb{N}^n$. Maka terdapat suatu himpunan berhingga dari vector-vektor $v_1, \dots, v_r \in S$ sedemikian sehingga

$$S \subseteq (v_1 + \mathbb{N}^n) \cup \dots \cup (v_r + \mathbb{N}^n).$$

2.2. Algoritme Pembagian pada Polinomial Multivariat

Penjelasan mengenai algoritme pembagian ini merujuk pada buku "*Concrete Abstract Algebra*" [8]. Algoritme pembagian pada polinomial multivariat dijelaskan sebagai berikut.

Algoritme 2.2.1: Algoritme pembagian pada polinomial multivariat.

INPUT: *Term ordering* $\leq$, polinomial $f \in R[X_1, \dots, X_n] \backslash \{0\}$, rangkaian polinomial tak nol $f_1, \dots, f_m \in R[X_1, \dots, X_n]$.
OUTPUT: Polinomial $a_1, \dots, a_m, r \in R[X_1, \dots, X_n]$ yang memenuhi $f = a_1 f_1 + \cdots + a_m f_m + r$.

1. *Set* $a_1 \leftarrow 0, \dots, a_m \leftarrow 0, r \leftarrow 0, s \leftarrow f$.
2. *While* $s \neq 0$:
   2.1. Jika $\text{in}_{\leq}(s)$ dapat dibagi dengan suatu $\text{in}_{\leq}(f_i)$, maka ambil i terkecil yang memenuhi, kemudian lakukan:
      (i) $s \leftarrow s - \frac{in_{\leq}(s)}{in_{\leq}(f_i)} f_i$.
      (ii) $a_i \leftarrow a_i + \frac{in_{\leq}(s)}{in_{\leq}(f_i)}$.
   2.2. Jika $\text{in}_{\leq}(s)$ tidak dapat dibagi dengan suatu $\text{in}_{\leq}(f_i)$, maka lakukan:
      (i) $r \leftarrow r + in_{\leq}(s)$.
      (ii) $s \leftarrow s - in_{\leq}(s)$.

Return $(a_1, \dots, a_m, r)$.

Definisi 2.2.2 *[8]*: Misalkan $f \in R[X_1, \dots, X_n]$ dan $F = (f_1, \dots, f_m)$ merupakan suatu rangkaian dari polinomial tak nol pada $R[X_1, \dots, X_n]$. Maka f^F menotasikan sisa pembagian r, hasil dari pembagian polinomial f oleh F menggunakan Algoritme 2.2.1: Algoritme pembagian pada polinomial multivariat.

2.3. Serangan Aljabar

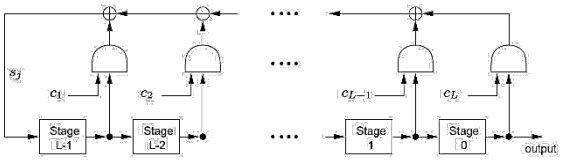
Serangan aljabar termasuk dalam kategori *known plaintext attack* [10]. Prinsip utama dari serangan aljabar adalah dengan mengubah permasalahan dalam menyerang sistem kriptografi (seperti menemukan *secret key*) menjadi permasalahan dalam menyelesaikan sistem persamaan polinomial. Serangan aljabar terdiri dari dua langkah. Langkah pertama, melakukan konversi algoritme penyandian yang mencakup beberapa informasi tambahan, seperti *s-box* dan *key schedule*, ke dalam bentuk sistem persamaan polinomial yang berhubungan dengan bit-bit kunci, bit-bit teks terang dan bit-bit teks sandi. Langkah kedua, menyelesaikan sistem persamaan polinomial dan mendapatkan solusi kunci dari algoritme tersebut. Untuk menyelesaikan sistem persamaan polinomial ini terdapat beberapa metode, antara lain basis Gröbner, *Linearization*, *Extended Linearization* (XL) *Algorithm*, dan ElimLin [3].

2.4. *Linear Feedback Shift Register (LFSR)*

Linear Feedback Shift Register (LFSR) merupakan salah satu komponen dasar yang banyak digunakan dalam algoritme *keystream generator* atau *stream cipher*. Hal ini disebabkan oleh beberapa faktor, baik dari sisi implementasi maupun keamanan [1].

Definisi 2.4.1 *[1]*: LFSR dengan panjang L terdiri atas L buah *stage*, yang diberi nomor $0,1, \dots, L-1$. Masing-masing *stage* dapat menyimpan satu bit data, dan memiliki satu masukan serta satu keluaran, selain itu setiap *stage* juga memiliki *clock* yang dapat mengatur pergerakan data. Untuk setiap satuan waktu, pada LFSR terdapat operasi berikut,

(i) isi pada *stage* ke-0 menghasilkan bagian dari rangkaian *keystream*,
(ii) isi pada *stage* ke-i dipindahkan ke dalam *stage* ke-$(i-1)$ untuk $1 \leq i \leq L-1$,
(iii) isi yang baru pada *stage* ke-$(L-1)$ merupakan nilai bit *feedback* s_j yang diperoleh melalui hasil operasi XOR dari isi dari *stage* tertentu.



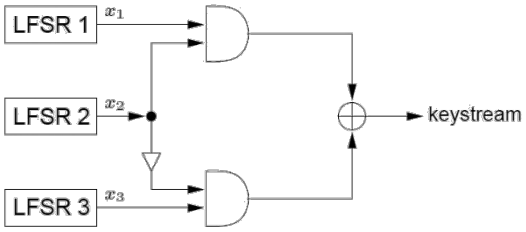
Gambar 1. *Linear Feedback Shift Register* (LFSR) dengan panjang *L [1]*.

Gambar 1 memperlihatkan LFSR berdasarkan Definisi 2.4.1. Setiap c_i bernilai 1 atau 0, terdapat gerbang logika AND, dan bit *feedback* s_j merupakan hasil operasi XOR dari *stage* ke-i untuk $0 \leq i \leq L-1$, dengan $c_{L-i} = 1$. Periode maksimal dari suatu LFSR dengan panjang L adalah $P = 2^L - 1$ [1].

Definisi 2.4.2 *[1]***:** LFSR pada Gambar 1 dinotasikan dengan $\langle L, C(D)\rangle$, dengan $C(D) = 1 + c_1 D + \cdots + c_L D^L \in \mathbb{Z}_2[D]$ merupakan fungsi polinomial.

2.5. *Geffe Generator*

Geffe generator merupakan salah satu *stream cipher* berbasis LFSR yang termasuk dalam kategori *nonlinear combination generators*. Skema *Geffe generator* ditunjukkan pada Gambar 2 [1].



Gambar 2. *Geffe Generator [1].*

Geffe generator terdiri atas tiga buah LFSR, yang dikombinasikan dengan fungsi kombinasi nonlinier yang berupa dua buah gerbang AND, satu buah gerbang NOT, dan satu buah gerbang XOR, atau dinotasikan dengan

$$f(x_1, x_2, x_3) = x_1 x_2 + (1 + x_2) x_3 = x_1 x_2 + x_2 x_3 + x_3.$$

3. METODOLOGI PENELITIAN

Pada penelitian ini digunakan metode kepustakaan dan ekperimen. Metode kepustakaan dilakukan dengan mempelajari dan memahami teori-teori terkait penelitian yang dilakukan. Sumber yang digunakan berupa buku, *paper*, tugas akhir, tesis, disertasi, jurnal, maupun sumber lainnya. Metode eksperimen dilakukan dengan menerapkan basis Gröbner untuk menyelesaikan sistem persamaan polinomial yang diperoleh dari serangan aljabar terhadap algoritme *Geffe generator*. Penerapan basis Gröbner untuk menyelesaikan sistem persamaan polinomial dilakukan menggunakan SageMath 8.7. Untuk eksperimen ini dipilih tiga buah LFSR sebagai komponen dari *Geffe generator*, yaitu LFSR_1, LFSR_2, dan LFSR_3, dengan panjang masing-masing adalah $L_1 = 3$, $L_2 = 2$, dan $L_3 = 4$.

Ada dua skenario yang digunakan dalam eksperimen ini. Pada skenario pertama, untuk fungsi *feedback* pada LFSR_1 dipilih sebuah fungsi polinomial primitif berderajat 3, untuk LFSR_2 dipilih sebuah fungsi polinomial primitif berderajat 2, sedangkan untuk LFSR_3 dipilih sebuah fungsi polinomial berderajat 4 namun bukan merupakan polinomial primitif. Pada skenario kedua, fungsi *feedback* untuk LFSR_1 dan LFSR_2 menggunakan fungsi polinomial yang sama dengan skenario pertama, namun untuk LFSR_3 dipilih sebuah fungsi polinomial primitif berderajat 4.

Pada skenario pertama dilakukan lima kali pembangkitan *keystream* dengan menggunakan lima buah *seed* berbeda yang dipilih secara acak oleh penulis. Pada skenario kedua juga dilakukan lima kali pembangkitan *keystream* dengan menggunakan lima buah *seed* yang sama yang digunakan pada skenario pertama. Hasilnya adalah diperoleh lima pasang *seed* dan *keystream* pada skenario pertama, dan lima pasang *seed* dan *keystream* pada skenario kedua.

Kemudian dilakukan pencarian sistem persamaan dari bit *keystream* dari keluaran *Geffe generator* dengan menentukan persamaan bit untuk setiap keluaran dari seluruh LFSR. Kemudian seluruh persamaan bit tersebut dimasukkan ke dalam fungsi *keystream* dari *Geffe generator* sehingga diperoleh sistem persamaan untuk seluruh *keystream*. Dari persamaan yang diperoleh kemudian ditambahkan dengan nilai bit masing-masing *keystream*. Sehingga diperoleh sistem persamaan bit yang merepresentasikan masing-masing *keystream*. Karena terdapat lima buah *keystream* untuk skenario satu dan lima buah *keystream* untuk skenario kedua, maka pada tahap ini diperoleh total 10 sistem persamaan.

Selanjutnya dibangun sebuah ideal dari masing-masing sistem persamaan yang diperoleh. Kemudian dilakukan pencarian basis Gröbner dari ideal tersebut. Apabila basis Gröbner berhasil diperoleh maka untuk menemukan solusi dari sistem persamaan bit dapat dilakukan dengan menyelesaikan sistem persamaan dari basis Gröbner tersebut. Kemudian solusi yang diperoleh dibandingkan dengan nilai *seed* yang berkorespondensi. Apabila solusi yang ditemukan bernilai ekuivalen dengan nilai *seed* maka serangan aljabar dengan metode penyelesaian menggunakan basis Gröbner berhasil dilakukan

4. KAJIAN DAN ANALISIS MATEMATIS BASIS GRÖBNER

Pembahasan mengenai kajian matematis basis Gröbner merujuk pada [8], jika disebutkan lain, akan dituliskan rujukannya.

4.1. Pengertian dan Sifat-sifat Basis Gröbner

Dalam pembahasan selanjutnya, diasumsikan terdapat suatu lapangan yang dinotasikan dengan k.

Definisi 4.1.1: Misalkan F himpunan polinomial tak nol.

$$F = (f_1, \ldots, f_m) \subseteq k[X_1, \ldots, X_n]$$

disebut sebagai basis Gröbner untuk ideal I dalam $k[X_1, \ldots, X_n]$ terhadap suatu *term ordering* $\leq$ jika $F \subseteq I$, dan untuk setiap $f \in I \backslash \{0\}$,

$$\text{in}_{\leq}(f_i) | \text{in}_{\leq}(f)$$

untuk suatu $i = 1, \dots, m$. Himpunan F disebut sebagai basis Gröbner terhadap suatu *term ordering* $\leq$ jika ia merupakan basis Gröbner untuk ideal $\langle f_1, \dots, f_m \rangle$ terhadap suatu *term ordering* $\leq$.

Perhatikan bahwa suatu ideal $I = \langle f_1, \dots, f_m \rangle$ merupakan ideal yang dibangun oleh kombinasi linier dari $(f_1, \dots f_m)$.

Proposisi 4.1.2: Misalkan $G = (f_1, \dots f_m)$ merupakan basis Gröbner terhadap term ordering $\leq$. Untuk suatu polinomial $f \in k[X_1, \dots X_n]$ berlaku

$$f \in I \Leftrightarrow f^G = 0,$$

dengan $I = \langle f_1, \dots, f_m \rangle$.

Bukti. Pembuktian dilakukan dengan dua tahap:
($\Leftarrow$) Jika $f^G = 0$ maka sesuai dengan algoritme pembagian dan Definisi 2.6.3, diperoleh $f = a_1 f_1 + \cdots + a_m f_m$ sehingga $f \in I = \langle f_1, \dots, f_m \rangle$;
($\Rightarrow$) Misalkan $f = a_1 f_1 + \cdots + a_m f_m + f^G$ merupakan hasil dari algoritme pembagian, dengan $r = f^G$. Maka

$$r = f - a_1 f_1 - \cdots - a_m f_m \in I.$$

Jika $r \neq 0$ maka terdapat suatu $\text{in}_{\leq}(f_i)$ yang habis membagi $\text{in}_{\leq}(r)$, karena $(f_1, \dots, f_m)$ diasumsikan sebagai basis Gröbner pada I. Pernyataan ini kontradiksi dengan fakta bahwa r merupakan sisa dari pembagian oleh G. Akibatnya $f^G = r = 0$. ■

Selanjutnya akan ditunjukkan bahwa setiap ideal di $k[X_1, \dots, X_n]$ memiliki basis Gröbner.

Teorema 4.1.3: Misalkan k adalah sebuah lapangan, notasi $\leq$ merupakan *term ordering*, dan $I \subseteq k[X_1, \dots, X_n]$ merupakan ideal. Maka ideal I memiliki basis Gröbner terhadap *term ordering* $\leq$.

Bukti. Misalkan $S = \{v \in \mathbb{N}^n | X^v = \text{in}_{\leq}(f)$ untuk suatu $f \in I\}$ dan $S \subseteq \mathbb{N}^n$. Sesuai lema Dickson (Lema 2.1.5), karena S himpunan bagian dari $\mathbb{N}^n$, maka secara berhingga terdapat $f_1, \dots, f_m \in I$ sedemikian sehingga

$$S \subseteq (v_1 + \mathbb{N}^n) \cup \dots \cup (v_m + \mathbb{N}^n),$$

dengan $X^{v_i} = \text{in}_{\leq}(f_i)$ untuk $i = 1, \dots, m$. Misalkan $aX^w = \text{in}_{\leq}(f)$, dengan $f \in I$. Maka $w = v_j + v$ untuk suatu $j = 1, \dots, m$ dan $v \in \mathbb{N}^n$. Ini berarti

$$X^w = X^{v_j + v}$$
$$X^w = X^{v_j} \cdot X^v.$$

Akibatnya $\text{in}_{\leq}(f_j) | \text{in}_{\leq}(f)$. Ini menyatakan bahwa $(f_1, \dots, f_m)$ merupakan basis Gröbner untuk ideal I. ■

4.2. Kriteria-S Buchberger

Terdapat kriteria tertentu untuk menentukan apakah suatu himpunan polinomial $F = (f_1, \dots, f_m)$ merupakan suatu basis Gröbner yang disebut dengan Kriteria-S Buchberger [8]. Pada pembahasan selanjutnya, notasi $\leq$ merupakan *term ordering* pada $k[X_1, \dots, X_n]$.

Definisi 4.2.1: Suatu $f \in k[X_1, \dots, X_n]$ dinyatakan tereduksi menuju nol modulo $F = (f_1, \dots, f_m) \subseteq R \backslash \{0\}$ jika terdapat $a_1, \dots, a_m \in k[X_1, \dots, X_n]$ sedemikian sehingga

$$f = a_1 f_1 + \cdots + a_m f_m$$

dan $\text{in}_{\leq}(a_i f_i) \leq \text{in}_{\leq}(f)$ jika $a_i f_i \neq 0$. Kondisi ini dinotasikan dengan

$$f \longrightarrow_F 0.$$

Proposisi 4.2.2: Misalkan $F = (f_1, \dots, f_m)$ dan $I = \langle f_1, \dots, f_m \rangle$.
(1) Jika $f \longrightarrow_F 0$ untuk setiap $f \in I$, maka F merupakan basis Gröbner untuk ideal I.
(2) Jika F merupakan basis Gröbner untuk ideal I, maka berlaku pernyataan bahwa $f^F = 0$ jika dan hanya jika $f \longrightarrow_F 0$ untuk setiap $f \in I$.

Bukti. Misalkan $f \in I \backslash \{0\}$.
(1) Perhatikan bahwa dapat diketahui jika $f \longrightarrow_F 0$ maka $\text{in}_{\leq}(f_j) | \text{in}_{\leq}(f)$ untuk suatu $f_j \in F$. Akibatnya jika $f \longrightarrow_F 0$ untuk setiap $f \in I$ maka F merupakan basis Gröbner untuk I.
(2) Akan dibuktikan dengan dua tahapan.
($\Rightarrow$) Pada Algoritme 2.2.1 diketahui bahwa $f^F = 0$ mengakibatkan $f \longrightarrow_F 0$.
($\Leftarrow$) Jika F merupakan basis Gröbner untuk I dan $f \longrightarrow_F 0$, maka $f^F = 0$ karena $f \in I$, ini sesuai dengan Proposisi 4.1.2.

Definisi 4.2.3: Polinomial-S dari dua polinomial f dan g terhadap suatu term ordering $\leq$, didefinisikan sebagai

$$S(f, g) = \frac{X^w}{\text{in}_{\leq}(f)} f - \frac{X^w}{\text{in}_{\leq}(g)} g,$$

dengan X^w merupakan KPK dari $\text{in}_{\leq}(f)$ dan $\text{in}_{\leq}(g)$.

Lema 4.2.4: Misalkan $F = (f_1, \dots, f_m)$ dan $I = \langle f_1, \dots, f_m \rangle$. Jika $S(f_i, f_j) \longrightarrow_F 0$ untuk setiap $i, j = 1, \dots, m$, maka $f \longrightarrow_F 0$ untuk setiap $f \in I$.

Berdasarkan penjelasan mengenai polinomial-S sebelumnya, maka terdapat suatu metode untuk memeriksa apakah suatu $F = (f_1, \dots, f_m)$ merupakan basis Gröbner. Metode ini disebut dengan kriteria-S Buchberger [8].

Teorema 4.2.5 (Buchberger): Suatu rangkaian polinomial $F = (f_1, \dots, f_m)$ merupakan basis Gröbner jika dan hanya jika $S(f_i, f_j) \longrightarrow_F 0$ untuk $1 \leq i < j \leq m$.

Bukti. Teorema ini merupakan konsekuensi langsung dari Proposisi 4.2.2 dan Lema 4.2.4.

Akibat 4.2.6: Suatu rangkaian polinomial $F = (f_1, \dots, f_m)$ merupakan basis Gröbner jika dan hanya jika $\left(S(f_i, f_j)\right)^F = 0$ untuk $1 \leq i < j \leq m$.

Bukti. Misalkan $F = (f_1, \dots f_m)$ dan $I = \langle f_1, \dots, f_m \rangle$. Pembuktian dilakukan dengan dua tahapan.

($\Leftarrow$) Jika $\left(S(f_i, f_j)\right)^F = 0$ untuk $1 \le i < j \le m$, maka berlaku $S(f_i, f_j) \rightarrow_F 0$ $1 \le i < j \le m$, dan sesuai dengan Teorema 4.2.5, maka F merupakan basis Gröbner.

($\Rightarrow$) Jika F merupakan basis Gröbner, dan karena $S(f_i, f_j) \in I$, maka sesuai Proposisi 4.1.2 berlaku $\left(S(f_i, f_j)\right)^F = 0$. ■

4.3. Algoritme Buchberger

Kriteria-S Buchberger dapat digunakan untuk memeriksa apakah suatu rangkaian polinomial $F = (f_1, \dots, f_m)$ merupakan basis Gröbner atau bukan. Berdasarkan kriteria tersebut, dikembangkan suatu algoritme untuk menemukan basis Gröbner pada suatu ideal I, yang disebut dengan algoritme Buchberger [8].

Algoritme 4.3.1: Algoritme Buchberger

INPUT: Rangkaian polinomial $F = (f_1, \dots, f_m)$, ideal $I = \langle f_1, \dots, f_m \rangle$, *term ordering* $\le$.
OUTPUT: Basis Gröbner G untuk ideal $I = \langle f_1, \dots, f_m \rangle$.

1. Untuk $1 \le i < j \le m$:
   1.1. Hitung $\left(S(f_i, f_j)\right)^F$.
   1.2. Jika $\left(S(f_i, f_j)\right)^F \neq 0$:
      (i) $F \leftarrow F \cup \left\{\left(S(f_i, f_j)\right)^F\right\}$.
      (ii) Kembali ke langkah 1.
2. *Return* $G \leftarrow F$.

Basis Gröbner $F = (f_1, \dots, f_m)$ untuk suatu ideal $I = \langle f_1, \dots, f_m \rangle$ tidak bersifat tunggal. Sebarang polinomial $f \in I$ dapat dimasukkan ke dalam F dan $F' = F \cup \{f\} = (f_1, \dots, f_m, f)$ tetap merupakan basis Gröbner untuk ideal I [8]. Oleh karena itu perlu diperoleh basis Gröbner yang bersifat tunggal.

Definisi 4.3.2: Basis Gröbner minimal $(f_1, \dots, f_m)$ merupakan basis Gröbner yang memenuhi sifat:

i. $\text{in}_{\le}(f_i)$ tidak dapat dibagi oleh $\text{in}_{\le}(f_j)$ untuk $i \neq j$,
ii. Koefisien dari $\text{in}_{\le}(f_i)$ adalah 1.

Basis Gröbner minimal juga tidak bersifat tunggal. Basis Gröbner yang bersifat tunggal adalah basis Gröbner tereduksi [8].

Definisi 4.3.3: Basis Gröbner tereduksi $(f_1, \dots, f_m)$ merupakan basis Gröbner minimal sedemikian sehingga tidak terdapat *term* pada f_i yang habis dibagi oleh $\text{in}_{\le}(f_j)$ untuk $i \neq j$.

4.4. Penyelesaian Sistem Persamaan Polinomial Multivariat Menggunakan Basis Gröbner

Misalkan diberikan sistem persamaan polinomial dalam n variabel pada suatu lapangan k sebagai berikut:

$$\begin{aligned} f_1(x_1, \dots x_n) &= 0, \\ f_2(x_1, \dots x_n) &= 0, \\ &\vdots \\ f_m(x_1, \dots x_n) &= 0. \end{aligned} \qquad (1)$$

Akan dicari solusi yang memenuhi sistem persamaan tersebut. Jika $n = 1$, maka diperoleh sistem persamaan polinomial dengan hanya satu variabel x_1. Sistem ini dapat diselesaikan menggunakan algoritme Euclid [8]. Perhatikan bahwa ideal $\langle f_1, \dots, f_m \rangle \subseteq k[x_1]$ yang dibangun oleh $f_1, \dots, f_m \in k[x_1]$ merupakan ideal utama $\langle f \rangle$ yang dibangun oleh f dengan $f = \text{FPB}(f_1, \dots, f_m)$. Polinomial f tersebut dapat dicari menggunakan algoritme Euclid. Karena $\langle f_1, \dots, f_m \rangle = \langle f \rangle$, maka $f_1(x) = \cdots = f_m(x) = 0$ jika dan hanya jika $f(x) = 0$. Sehingga proses penyelesaian sistem tersebut telah direduksi sehingga cukup dengan menyelesaikan satu persamaan.

Misalkan $V(f_1, \dots, f_m)$ merupakan himpunan penyelesaian untuk sistem persamaan (1) atau dinotasikan dengan
$\{(a_1, \dots, a_n) \in k^n \mid f_i(a_1, \dots, a_n) = 0, \forall i = 1, \dots, m\}$.
Perhatikan bahwa himpunan $V(f_1, \dots, f_m)$ juga merupakan himpunan yang sama dengan $V(I)$ yang dinotasikan

$$\{(a_1, \dots, a_n) \in k^n \mid f(a_1, \dots, a_n) = 0 \; \forall f \in I\},$$

dengan $I = \langle f_1, \dots, f_m \rangle$ [8]. Hal ini disebabkan ideal I merepresentasikan seluruh persamaan yang dapat diperoleh melalui kombinasi linier dari $f_1, \dots, f_m$. Dalam kondisi tertentu, jika diketahui basis Gröbner $(g_1, \dots, g_r)$ untuk ideal I, maka $V(g_1, \dots, g_r)$ juga merupakan solusi untuk persamaan-persamaan pada I. Sehingga diperoleh

$$V(f_1, \dots, f_m) = V(g_1, \dots, g_r).$$

Intinya adalah bahwa sistem persamaan

$$\begin{aligned} g_1(x_1, \dots x_n) &= 0, \\ g_2(x_1, \dots x_n) &= 0, \\ &\vdots \\ g_r(x_1, \dots x_n) &= 0 \end{aligned}$$

sering kali lebih mudah diselesaikan dibandingkan dengan sistem persamaan (1) [8].

Contoh 4.4.1: Diberikan sistem persamaan berikut,

$$\begin{aligned} Y^2 - X^3 + X &= 0, \\ Y^3 - X^2 &= 0 \end{aligned} \qquad (2)$$

dalam $\mathbb{R}^2$. Solusi dari sistem persamaan ini akan dicari menggunakan basis Gröbner.

Untuk menyelesaikan sistem persamaan menggunakan basis Gröbner, sistem persamaan (2) terlebih dahulu diubah ke dalam bentuk sistem persamaan lain. Pertama didefinisikan ideal $I = \langle Y^2 - X^3 + X, Y^3 - X^2 \rangle$ dan *lexicographic ordering* $\le$ dengan $X \ge Y$. Selanjutnya dilakukan pencarian basis Gröbner untuk ideal I berdasarkan *term*

ordering $\leq$. Proses ini dilakukan dengan dua metode, yaitu menggunakan perhitungan manual dan menggunakan aplikasi SageMath.

1. Proses perhitungan manual
   (i) Proses pencarian dilakukan menggunakan algoritme Buchberger, dan diperoleh
$(-X^3 + X + Y^2, -X^2 + Y^3, XY^3 - X - Y^2, XY^2 + Y^3 - Y^6, Y^9 - 2Y^6 - Y^4 + Y^3, -X - Y^2 - Y^4 + Y^7)$.
   (ii) Basis Gröbner tersebut kemudian direduksi sehingga diperoleh basis Gröbner tereduksi
$G = (Y^9 - 2Y^6 - Y^4 + Y^3, X + Y^2 + Y^4 - Y^7)$.

2. Proses dengan aplikasi SageMath
   (i) Pada aplikasi SageMath, pencarian basis Gröbner dapat dilakukan dengan menggunakan perintah groebner_basis. Kode sumber untuk proses ini terdapat pada Lampiran 1. Perintah groebner_basis pada SageMath menghasilkan keluaran berupa basis Gröbner tereduksi, dan diperoleh

$$G = (Y^9 - 2Y^6 - Y^4 + Y^3, X + Y^2 + Y^4 - Y^7).$$

Dari kedua metode di atas diperoleh basis Gröbner yang sama sehingga solusi untuk sistem persamaan (2) dapat diperoleh dengan menyelesaikan sistem persamaan

$$\begin{aligned} g_1 &= Y^9 - 2Y^6 - Y^4 + Y^3 = 0, \\ g_2 &= X + Y^2 + Y^4 - Y^7 = 0. \end{aligned} \quad (3)$$

Selanjutnya dilakukan penyelesaian sistem persamaan (3). Perhatikan bahwa terdapat persamaan yang hanya mengandung satu variabel, yaitu g_1 dengan hanya mengandung variabel Y. Sehingga dapat terlebih dahulu diselesaikan persamaan $g_1 = 0$. Dengan menggunakan aplikasi SageMath diperoleh nilai Y, yaitu $Y = 0, Y = 0.605423$, dan $Y = 1.2876$. Selanjutnya nilai-nilai tersebut disubstitusikan ke dalam g_2 sehingga diperoleh solusi sistem persamaan (2) adalah $(0, 0)$, $(-0.471073, 0.605423)$, dan $(1.46109, 1.2876)$.

5. PENERAPAN BASIS GRÖBNER DALAM MENYELESAIKAN SERANGAN ALJABAR

Penyelesaian sistem persamaan polinomial multivariat pada penelitian ini dilakukan dengan menerapkan basis Gröbner. Sistem persamaan yang digunakan merupakan hasil serangan aljabar terhadap algoritme *Geffe generator*. Proses komputasi pada eksperimen ini menggunakan aplikasi Microsoft Excel 2013 dan SageMath 8.7.

Dalam eksperimen ini terdapat tiga buah LFSR yang menjadi komponen dasar dari *Geffe generator*, yaitu LFSR_1, LFSR_2, dan LFSR_3. LFSR_1 berukuran $L = 3$, dengan *internal state* (a_3, a_2, a_1). LFSR_2 berukuran $L = 2$, dengan *internal state* (b_2, b_1). LFSR_3 berukuran $L = 4$, dengan *initial state* (c_4, c_3, c_2, c_1). Ketiga LFSR tersebut dikombinasikan ke dalam skema *Geffe generator* sehingga diperoleh fungsi *keystream*

$$f(a_i, b_i, c_i) = a_i b_i + (1 + b_i)c_i = a_i b_i + b_i c_i + c_i,$$

dengan a_i merepresentasikan keluaran dari LFSR_1, b_i merepresentasikan keluaran dari LFSR_2, dan c_i merepresentasikan keluaran dari LFSR_3.

Pada eksperimen dilakukan pembangkitan lima buah *keystream* dengan lima buah *seed* berbeda. Tabel 1 menunjukkan daftar *seed* yang digunakan.

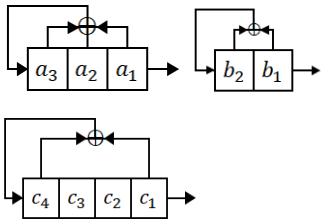
Tabel 1. Daftar *seed* yang digunakan.

No	*Seed*								
	a_3	a_2	a_1	b_2	b_1	c_4	c_3	c_2	c_1
1	1	1	0	0	1	0	1	1	0
2	0	1	1	1	1	1	1	0	0
3	1	1	1	1	1	1	1	1	1
4	1	0	1	0	1	1	0	0	1
5	0	1	0	1	0	0	0	1	0

Pada eksperimen ini dipilih fungsi *feedback* f_1, f_2, dan f_3 secara berturut-turut untuk LFSR_1, LFSR_2, dan LFSR_3 sebagai berikut

$$\begin{aligned} f_1 &= x^3 + x + 1, \\ f_2 &= x^2 + x + 1, \\ f_3 &= x^4 + x + 1. \end{aligned}$$

Internal state pada LFSR_1 dinotasikan dengan (a_1, a_2, a_3), pada LFSR_2 dinotasikan dengan (b_1, b_2), dan pada LFSR_3 dinotasikan dengan (c_1, c_2, c_3, c_4). Skema fungsi *feedback* pada ketiga LFSR ditunjukkan dalam Gambar 3. Dengan fungsi *feedback* tersebut, diperoleh LFSR_1 memiliki periode sebanyak tujuh, LFSR_2 memiliki periode sebanyak tiga, dan LFSR_3 memiliki periode sebanyak 15. Oleh karena itu, dengan ketiga LFSR tersebut diperoleh *Geffe generator* dengan periode $\text{KPK}(7,3,15) = 105$.



Gambar 3. Skema LFSR yang digunakan pada eksperimen.

Selanjutnya dilakukan pembangkitan *keystream* dengan memasukkan *seed* ke dalam algoritme *Geffe generator*. Karena diketahui periode dari *Geffe generator* adalah 105, maka dilakukan pembangkitan *keystream* sepanjang 105 bit. Tabel 2 menunjukkan daftar *keystream* yang diperoleh. Sehingga diperoleh lima pasangan *seed* dan *keystream*.

Tabel 2. *Keystream* yang diperoleh pada eksperimen.

No	*Keystream*
1	0111 0100 1110 1001 1100 0001 1110 0111 0100 1010 1000 1111 0011 0011 0101 0100 0110 1011 1001 1010 1000 0111 0101 1100 1001 1100 0
2	1111 0111 0010 0101 0100 1110 1001 1001 1011 1000 0011 0101 1110 1001 0100 0011 1110 0110 0100 1110 1000 1101 0011 1011 0101 0000 0
3	1110 1101 1101 0001 1110 1111 0110 1010 1001 1111 0111 0011 0111 0100 1110 1011 1101 1011 1000 0111 1101 1110 1001 1101 0011 1110 0
4	1010 0111 0110 1010 0000 1111 0111 0010 0101 0100 1110 1001 1001 1011 1000 0011 0101 1110 1001 0100 0011 1110 0110 0100 1110 1000 1
5	0100 1110 1001 1100 0001 1110 0111 0100 1010 1000 1111 0011 0011 0101 0100 0110 1011 1001 1010 1000 0111 0101 1100 1001 1100 0011 1

Selanjutnya dilakukan serangan aljabar terhadap kelima buah *keystream* yang diperoleh. Pertama ditentukan persamaan bit untuk setiap keluaran dari ketiga LFSR, sebagaimana ditunjukkan pada Tabel 3.

Tabel 3. Rangkaian keluaran LFSR

LFSR_1	
t	a_i
1	a_1
2	a_2
3	a_3
4	$a_1 + a_3$
5	$a_1 + a_2 + a_3$
6	$a_1 + a_2$
7	$a_2 + a_3$

LFSR_2	
t	b_i
1	b_1
2	b_2
3	$b_1 + b_2$

LFSR_3	
t	c_i
1	c_1
2	c_2
3	c_3
4	c_4
5	$c_1 + c_4$
6	$c_1 + c_2 + c_4$
7	$c_1 + c_2 + c_3 + c_4$
8	$c_1 + c_2 + c_3$
9	$c_2 + c_3 + c_4$
10	$c_1 + c_3$
11	$c_2 + c_4$
12	$c_1 + c_3 + c_4$
13	$c_1 + c_2$
14	$c_2 + c_3$
15	$c_3 + c_4$

Dengan memasukkan semua nilai a_i, b_i, dan c_i ke dalam fungsi *keystream Geffe generator*, maka dapat diperoleh seluruh fungsi keluaran untuk setiap bit *keystream* dari skema *Geffe generator*.

$$\begin{aligned} w_1 &= a_1b_1 + b_1c_1 + c_1, \\ w_2 &= a_2b_2 + b_2c_2 + c_2, \\ w_3 &= a_3(b_1 + b_2) + (b_1 + b_2)c_3 + c_3 \\ &= a_3b_1 + a_3b_2 + b_1c_3 + b_2c_3 + c_3, \\ w_4 &= (a_1 + a_3)b_1 + b_1c_4 + c_4 \\ &= a_1b_1 + a_3b_1 + b_1c_4 + c_4. \\ &\vdots \end{aligned}$$

Nilai-nilai *keystream* pada Tabel 2 dimasukkan ke dalam fungsi keluaran tersebut, sehingga diperoleh sistem persamaan bit untuk setiap *keystream*. Misalkan diambil *keystream* pertama, maka diperoleh sistem persamaan

$$\begin{aligned} z_1^1 &= w_1 + 0 = a_1b_1 + b_1c_1 + c_1, \\ z_2^1 &= w_2 + 1 = a_2b_2 + b_2c_2 + c_2 + 1, \\ z_3^1 &= w_3 + 1 = a_3b_1 + a_3b_2 + b_1c_3 + b_2c_3 + c_3 + 1, \\ z_4^1 &= w_4 + 1 = a_1b_1 + a_3b_1 + b_1c_4 + c_4 + 1. \\ &\vdots \end{aligned}$$

Untuk melakukan pencarian solusi dibangun lima buah ideal yang berkorespondensi dengan masing-masing *keystream*, yaitu

$$\begin{aligned} I_1 &= \langle z_1^1, z_2^1, \dots, z_{105}^1 \rangle, \\ I_2 &= \langle z_1^2, z_2^2, \dots, z_{105}^2 \rangle, \\ I_3 &= \langle z_1^3, z_2^3, \dots, z_{105}^3 \rangle, \\ I_4 &= \langle z_1^4, z_2^4, \dots, z_{105}^4 \rangle, \\ I_5 &= \langle z_1^5, z_2^5, \dots, z_{105}^5 \rangle. \end{aligned}$$

Dari masing-masing ideal tersebut kemudian dicari lima basis Gröbner terhadap *lexicographic ordering* yang juga berkorespondensi dengan masing-masing *keystream*, yaitu

$$\begin{aligned} G_1 &= (a_1, a_2 + 1, a_3 + 1, b_1 + 1, b_2, c_1, c_2 + 1, c_3 + 1, c_4), \\ G_2 &= (a_1 + 1, a_2 + 1, a_3, b_1 + 1, b_2 + 1, c_1, c_2, c_3 + \\ &\quad 1, c_4 + 1), \\ G_3 &= (a_1 + 1, a_2 + 1, a_3 + 1, b_1 + 1, b_2 + 1, c_1 + 1, c_2 + \\ &\quad 1, c_3 + 1, c_4 + 1), \\ G_4 &= (a_1 + 1, a_2, a_3 + 1, b_1 + 1, b_2, c_1 + 1, c_2, c_3, c_4 + 1), \\ G_5 &= (a_1, a_2 + 1, a_3, b_1, b_2 + 1, c_1, c_2 + 1, c_3, c_4). \end{aligned}$$

Pencarian solusi untuk seluruh sistem persamaan bit dapat diperoleh dengan menyelesaikan sistem persamaan dari basis Gröbner-nya. Seluruh solusi yang diperoleh ditunjukkan pada Tabel 4.

Tabel 4. Perbandingan solusi yang diperoleh dengan nilai *seed*

No	*Seed*	Solusi
1	110 01 0110	110 01 0110
2	011 11 1100	011 11 1100
3	111 11 1111	111 11 1111
4	101 01 1001	101 01 1001
5	010 10 0010	010 10 0010

Dari Tabel 4 dapat terlihat bahwa seluruh solusi yang diperoleh ekuivalen dengan nilai *seed* yang berkorespondensi. Sehingga serangan aljabar pada algoritme *Geffe* dengan metode penyelesaian menggunakan basis Gröbner berhasil dilakukan.

6. SIMPULAN DAN SARAN

Basis Gröbner dapat diterapkan dalam menyelesaikan sistem persamaan polinomial multivariat yang diperoleh dari serangan aljabar. Hal ini dapat diamati baik dari teori yang telah dikaji, maupun dari eksperimen yang telah dilakukan. Dalam eksperimen yang dilakukan, sistem persamaan polinomial berhasil direduksi. Hal ini menyebabkan proses pencarian solusi menjadi lebih mudah.

Pada eksperimen yang dilakukan, ideal dibangun dengan menggunakan seluruh polinomial yang diperoleh. Alangkah baiknya jika ada suatu metode untuk membangun ideal yang sama dengan hanya mengambil beberapa polinomial saja dari seluruh polinomial tersebut. Denagn demikian komputasi dalam pencarian basis Gröbner menjadi lebih ringan sehingga pencarian basis Gröbner untuk sistem persamaan yang lebih besar dapat dilakukan.

Referensi

[1] A. J. Menezes, P. C. van Oorschot dan S. A. Vanstone, Handbook of Applied Cryptography, 1996.

[2] B. Schneier, Applied Cryptography, 2nd penyunt., John Wiley and Sons Inc., 1996.

[3] G. V. Bard, Algebraic Cryptanalysis, Springer Science & Business Media, 2009.

[4] S. Simmons, “Algebraic Cryptanalisis of Simplified AES,” pp. 305-314, 2009.

[5] B. Buchberger, “Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal,” *Ph.D. thesis, Innsbruck*, 1965.

[6] B. Buchberger, “Bruno Buchberger’s PhD thesis 1965: An Algorithm for Finding the Basis Elements of the Residue Class Ring of a Zero Dimensional Polynomial Ideal,” 2006.

[7] I. A. Ajwa, Z. Liu dan P. S. Wang, “Gröbner Bases Algorithm,” *Technical Reports of the Institute for Computational Mathematics*, 1995.

[8] N. Lauritzen, Concrete Abstract Algebra: From Numbers to Gröbner Bases, Cambridge University Press, 2003.

[9] V. Ene dan J. Herzog, Gröbner Bases in Commutative Algebra, American Mathematical Society, 2012.

[10] F. Paradise, “Serangan Aljabar pada Simplified Data Encryption Standard (S-DES) dengan Metode XL Algorithm,” Sekolah Tinggi Sandi Negara, 2018.

Perancangan Program Pelatihan Teknis Berdasarkan Kompetensi NIST SP 800-181 *National Initiative for Cybersecurity Education* (NICE) *Cybersecurity Workforce Framework* bagi Perespons Insiden Keamanan Siber Lembaga XYZ

Dicky Yudha Bimantara[1)] dan Kholif Faiz Ma'ruf[2)]
(1) *Politeknik Siber dan Sandi Negara / dicky.yudha@student.stsn-nci.ac.id*
(2) *Badan Siber dan Sandi Negara / kholif.faiz@bssn.go.id*

Abstrak

Transformasi dari suatu organisasi menjadi sebuah organisasi baru mengakibatkan perubahan standar kompetensi SDM sesuai dengan tupoksi baru tersebut yang berakibat munculnya permasalahan kesenjangan kompetensi pada pegawai. Hasil Analisis Kebutuhan Diklat (AKD) terhadap Perespons Insiden Keamanan Siber XYZ menyatakan bahwa terjadi kesenjangan antara kompetensi yang dimiliki sekarang dengan kompetensi ideal yang terdapat pada NICE Framework. Salah satu upaya pengembangan kompetensi pegawai adalah dengan diberikannya program pelatihan. Penelitian ini merancang suatu program pelatihan teknis yang disesuaikan dengan hasil AKD dari Tim Perespons Insiden Keamanan Siber XYZ untuk mengurangi kesenjangan kompetensi yang ada. Perancangan program pelatihan ini dibuat dengan mengikuti langkah analyze dan design yang merupakan tahapan dari kerangka Model ADDIE. AKD menghasilkan 10 kompetensi yang menjadi mayoritas senjang. Kemudian dilakukan pencarian terhadap kompetensi yang paling dibutuhkan untuk dibuatkan program pelatihan dengan menggunakan Metode Urgency, Seriousness, Growth (USG) yang menghasilkan 5 kompetensi dengan kebutuhan tertinggi. Program pelatihan yang dihasilkan bernama Pelatihan Respons Insiden dan Forensik Digital yang memiliki 5 Rancang Bangun Pembelajaran Mata Pelatihan (RBPMP) untuk memenuhi tujuan pelatih an tersebut. Penelitian ini juga menghasilkan training path bagi Perespons Insiden Keamanan Siber dalam mendukung pengembangan kariernya.

Kata Kunci: *AKD* (1), *NICE Framework* (2), *Model ADDIE* (3), *Metode USG* (4), *Program Pelatihan* (5), *Training Path* (6).

1. PENDAHULUAN

SDM yang berkompeten dan berkualitas dapat diperoleh dengan melakukan pengembangan terhadap SDM tersebut. Peraturan Kepala Lembaga Administrasi Negara (LAN) Nomor 10 Tahun 2018 menyatakan bahwa bentuk pengembangan kompetensi SDM salah satunya adalah melalui kegiatan pelatihan. Pelatihan merupakan salah satu komponen penting dalam pengembangan SDM pada sebuah institusi [1]. Belum adanya standar kompetensi yang sesuai dengan tugas pokok dan fungsi (tupoksi) keamanan siber menjadikan program pelatihan yang dijalankan dikhawatirkan tidak sesuai dengan kebutuhan.

Penggunaan NICE *Framework* sebagai pedoman standar kompetensi Perespons Insiden Keamanan Siber disebabkan karena belum adanya standar kompetensi terkait keamanan siber di lembaga terkait. Alasan lain penggunaan NICE *Framework* karena standar ini merupakan dasar bagi Lembaga XYZ dalam membentuk struktur dan tugas dari setiap susunan yang ada pada tubuh organisasi yang baru.Terdapat kesenjangan sebesar 45,34% pada Deputi terkait yang menandakan perlu adanya pengembangan dan peningkatan kompetensi SDM terkait [2].

Pusat Pendidikan dan Pelatihan (Pusdiklat) XYZ memiliki tugas melaksanakan penyelenggaraan pelatihan SDM keamanan siber dan sandi dan akreditasi lembaga pelatihan serta evaluasi dan pelaporan. Berdasarkan hasil observasi di Pusdiklat XYZ, diperoleh fakta tentang kebutuhan Pusdiklat untuk menyusun program pelatihan yang sesuai dengan standar kompetensi dan kebutuhan organisasi. Pelatihan dipilih karena memiliki peran strategis untuk meningkatkan kualitas SDM yang profesional baik dalam hal kompetensi, sikap, dan perilaku yang diharapkan sesuai dengan tugas dan peranannya masing-masing [3].

Salah satu model dalam perancangan program pelatihan adalah Model ADDIE, yang merupakan singkatan dari *Analyze*, *Design*, *Development*, *Implementation*, dan *Evaluation.* Model tersebut cocok untuk mengembangkan produk model instruksional/pembelajaran yang tepat sasaran, efektif, dan dinamis serta sangat membantu dalam pengembangan pembelajaran bagi penyelenggara program pelatihan [1]. Perancangan program pelatihan dihasilkan pada tahap *Design*. Model ADDIE dapat mempermudah pengajar dalam merencanakan pembelajaran yang berkualitas, efektif, dan efisien [4].

Berdasarkan dengan permasalahan yang telah diuraikan sebelumnya, penelitian ini bertujuan menyelesaikan permasalahan tentang kesenjangan kompetensi pada Perespons Insiden Keamanan Siber Lembaga XYZ. Caranya adalah dengan merancang program pelatihan teknis berdasarkan Kompetensi NIST SP 800-181 *National Inititatice for Cybersecurity Education* (NICE) *Cybersecurity Workforce Framework.*

2. LANDASAN TEORI

2.1. NICE *Framework*

NICE *Framework* berisi tentang pedoman yang menjelaskan ruang lingkup interdisipliner pada pekerjaan *cybersecurity* dan dapat dijadikan referensi dalam menggambarkan dan memberikan informasi terkait pekerjaan *cybersecurity* dari *knowledge*, *skill*, *abilities*, dan *tasks* yang diperlukan untuk memenuhi tugas *cybersecurity* suatu organisasi [2]. Lima fungsi dasar yang terdapat pada NIST *cybersecurity Framework* erat kaitannya dengan NICE *Framework* yakni bahwa kedua *framework* tersebut saling melengkapi dan apabila diterapkan dapat memberikan pendekatan yang jelas dalam mencapai tujuan *cybersecurity* baik dalam lingkup struktur organisasi maupun SDM organisasi [6].

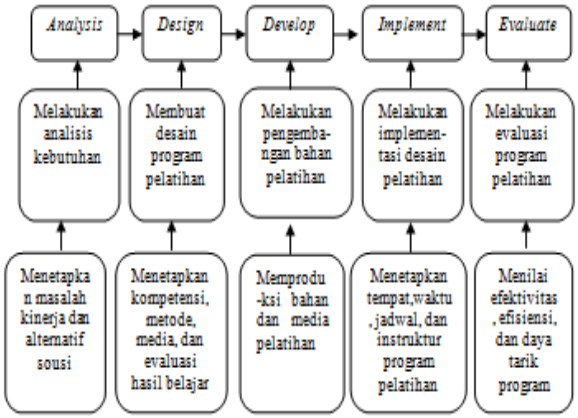
Penggunaan NICE *Framework* ditujukan sebagai upaya pengembangan tenaga kerja, tujuan pendidikan, dan atau pelatihan dengan mekanisme ketika materi yang digunakan di tingkat organisasi, maka pengguna harus menyesuaikan apa yang ada di NICE *Framework* mulai dari standar, regulasi, kebutuhan, dan misi organisasi pengguna [6]. NICE *Framework* merupakan dasar yang akan memperkuat kemampuan organisasi untuk berrkembang secara konsisten dan jelas tentang pekerjaan keamanan siber. Organisasi dapat menambah dan mengembangkannya untuk memenuhi kebutuhan dalam menyediakan panduan tentang berbagai aspek mengenai pengembangan, perencanaan, pelatihan, dan pendidikan pegawai keamanan siber dalam berbagai aspek [6].

2.2. Perespons Insiden Keamanan Siber

Perespons insiden keamanan siber memiliki tugas untuk melakukan investigasi, analisis, dan respons terhadap insiden siber yang terjadi di jaringan internet [6]. Perespons Insiden Keamanan Siber adalah jabatan yang melakukan tugas investigasi, analisis, dan respons terhadap insiden siber yang terjadi di jaringan internet Insiden berarti kejadian tak terduga yang menyebabkan gangguan operasi normal sedangkan insiden keamanan berarti suatu peristiwa yang menghalangi/mengganggu akses yang sah terhadap sistem atau informasi juga bisa diartikan sebagai suatu kejadian pelanggaran terhadap kebijakan keamanan.

2.3. Model ADDIE

Model ADDIE merupakan salah satu model sistem pembelajaran yang dapat diimplementasikan untuk mendesain dan mengembangkan program pelatihan yang efektif dan efisien [1]. Model ini mencerminkan atau menggambarkan adanya sejumlah langkah dan prosedur yang sistematik dan sistemik untuk digunakan dalam mencapai sasaran yang diinginkan. Salah satu fungsi ADDIE yaitu menjadi pedoman dalam membangun perangkat dan infrastruktur program pelatihan yang efektif, dinamis, dan mendukung kinerja pelatihan itu sendiri [1]. Secara ringkas Model ADDIE yang digunakan dalam mendesain dan mengembangkan sebuah program pelatihan dapat dilihat dalam ilustrasi berikut ini:



Gambar 1. Proese Model ADDIE [1]

Pada penelitian ini, tahapan yang digunakan adalah tahap *analyze* dan *design*, karena hasil akhir yang diharapkan adalah terciptanya suatu rancangan program pelatihan yang sesuai hasil AKD.

2.4. Metode USG

Urgency, *Seriousness*, *Growth* (USG) adalah salah satu *tools* untuk menyusun urutan prioritas isu yang harus diselesaikan. Caranya dengan menentukan tingkat urgensi, keseriusan, dan perkembangan isu dengan menentukan skala nilai 1-5. Isu yang memiliki total skor tertinggi merupakan isu prioritas [1]. Metode USG merupakan salah satu cara menetapkan urutan prioritas masalah dengan metode teknik *scoring*. Proses untuk metode USG dilaksanakan dengan memperhatikan urgensi dari masalah, keseriusan masalah yang dihadapi, serta kemungkinan berkembangnya masalah tersebut semakin besar.

2.5. Metode Delphi

Metode Delphi dapat dikarakteristikkan sebagai sebuah metode berkelompok untuk menstrukturkan hal dengan proses komunikasi sehingga proses akan menjadi efektif untuk menyelesaikan permasalahan yang kompleks [7]. Metode delphi merupakan suatu metode dimana dalam proses pengambilan keputusan melibatkan beberapa pakar [2]. Adapun para pakar tersebut tidak dipertemukan secara langsung (tatap muka), dan identitas dari masing-masing pakar disembunyikan sehingga setiap pakar tidak mengetahui identitas pakar yang lain. Hal ini bertujuan untuk menghindari adanya dominasi pakar lain dan dapat meminimalkan pendapat yang bias.

Metode Delphi ini telah diaplikasikan dalam hal pengambilan kebijakan, perencanaan, atau ide yang berdasarkan pada pemikiran atau *judgement*. Metode ini sangat berguna untuk mengumpulkan pendapat dan *judgement* para ahli dan praktisi ketika waktu dan jarak serta faktor lain tidak memungkinkan bagi

mereka bertemu dalam satu lokasi yang sama [7]. Terdapat 4 tahapan penting dalam Metode Delphi (RAND), yaitu:

a. Eksplorasi pendapat
b. Merangkum pendapat para pakar dan mengkomunikasikannya kembali
c. Mencari informasi mengenai alasan para pakar terkait pendapat yang disampaikan
d. Revisi pendapat pada tahap hasil
e. Evaluasi

2.6. Taksonomi Bloom

Benjamin Bloom dan kawan-kawan (1956) mengemukakan taksonomi kemampuan yang mencakup aspek kognitif atau pengetahuan, aspek afektif atau sikap, dan aspek psikomotor atau keterampilan [8]. Aspek-aspek tersebut pada hakikatnya merupakan suatu kesatuan yang utuh yang dimiliki oleh seseorang. Aspek kognitif terkait dengan kemampuan intelektual atau kemampuan seseorang dalam mempelajari ilmu pengetahuan.

Bloom dan kawan-kawan (1956) mengemukakan 6 kemampuan yang bersifat hierarkis yang terdapat pada aspek kognitif [8]:

a. Pengetahuan: kemampuan dalam mengidentifikasi dan menyebutkan informasi dan data faktual.
b. Pemahaman: kemampuan dalam menjelaskan dan mengartikan suatu konsep.
c. Aplikasi: kemampuan dalam menerapkan prinsip dan aturan yang telah dipelajari sebelumnya.
d. Analisis: kemampuan menguraikan sebuah konsep dan menjelaskan saling keterkaitan komponen-komponen yang terdapat di dalamnya.
e. Sintesis: kemampuan untuk menggabungkan komponen-komponen menjadi sebuah konsep atau aturan yang baru.
f. Evaluasi: kemampuan dalam menilai objek dan membuat keputusan terhadap sebuah situasi yang dihadapi.

3. METODOLOGI PENELITIAN

Penelitian ini menggunakan *mix method* dengan mengkombinasikan dua metode penelitian sekaligus yakni kualitatif dan kuantitatif dalam suatu kegiatan penelitian, sehingga akan diperoleh data yang lebih komprehensif, valid, reliabel, dan objektif [9]. Langkah pertama yang dilakukan adalah tahap *analyze* yakni melakukan AKD. Hasil akhir dari proses AKD ini adalah standar kompetensi dari kelembagaan yang dimaksud. Tahapan selanjutnya adalah melakukan analisis *gap*/ kesenjangan pada kompetensi yang dimiliki oleh pegawai saat ini dengan kompetensi yang seharusnya dimiliki. Analisis kesenjangan yang dihasilkan dari proses peta kompetensi akan memberikan gambaran mengenai keadaan SDM pada suatu organisasi.

Kemudian melakukan analisis prioritas kompetensi yang akan dibuatkan program pelatihan dikarenakan keterbatasan waktu dan sumber daya dalam proses pengerjaan tugas akhir yang tidak memungkinkan semua kompetensi dibuatkan program pelatihan. Pengambilan kompetensi prioritas menggunakan Model USG (*Urgency*, *Seriousness*, dan *Growth*), lalu masuk ke tahapan *design* yakni membuat rancangan program pelatihan yang disusun berdasarkan prinsip SMART (*Specific, Measurable, Attainable, Realistic, Timely*). Prinsip ini tidak hanya digunakan untuk merumuskan tujuan pembelajaran saja, namun dalam perumusan tujuan lain seperti tujuan program, pendirian lembaga, visi misi juga menggunakan prinsip ini [10].

Proses selanjutnya menggunakan Metode Delphi yang diawali dengan menentukan Pakar/ Narasumber yang akan dijadikan *panel expert* untuk dimintai pendapat [2]. Tahapan iterasi pada *round* Delphi menggunakan wawancara yang dibantu dengan daftar kompetensi bagi SDM XYZ berdasarkan NICE *Framework.* Dokumen pelengkap lainnya diperoleh dari pedoman-pedoman atau peraturan yang mengatur tentang proses perancangan program pelatihan yang dikeluarkan oleh Lembaga Administrasi Negara (LAN) dan Lembaga Negara lainnya. Dalam hal ini proses yang dilakukan adalah iterasi ke-1 dimana para pakar memberikan tanggapan yang kemudian hasilnya ditabulasi untuk dilakukan analisis. Apabila hasil isian kuesioner dari tiap pakar ada perbedaan, maka hasil tersebut akan direkap, dianalisis, dimodifikasi, dan disusun menjadi daftar isian program pelatihan baru untuk kemudian diberikan kepada pakar [2]. Proses iterasi seperti ini dilakukan hingga iterasi *round* ke-3, namun apabila telah didapatkan kesepakatan dari para pakar maka kesepakatan tersebut yang akan diambil sebagai program pelatihan yang paling tepat pada SDM Lembaga XYZ berdasarkan variabel kompetensi *knowledge, skill, abilities,* dan *task*.

4. PERANCANGAN PROGRAM PELATIHAN

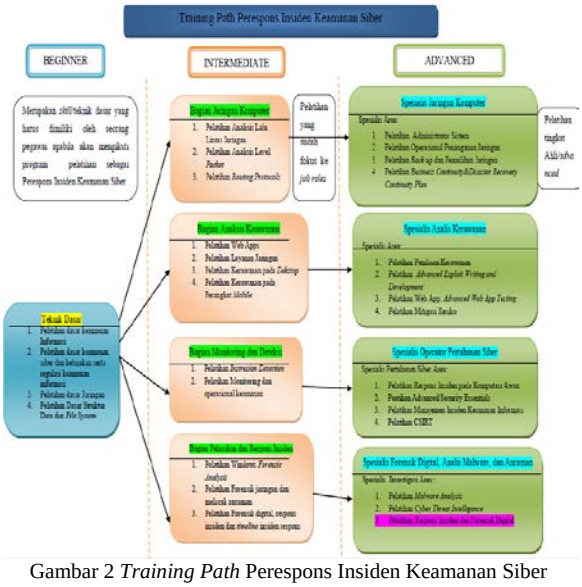
4.1. Perancangan *Training Path* Perespons Insiden Keamanan Siber

Training Path dirancang berdasarkan kompetensi ideal yang terdapat pada NICE *Framework*. Perancangan ini dilakukan dengan memetakan kompetensi Persespons Insiden Keamanan Siber di NICE *Framework* ke dalam 3 kelompok kategori pelatihan. *Training Path* ini nantinya akan terdiri dari kelompok-kelompok pelatihan yang diklasifikasikan berdasarkan tingkatannya menurut NIST SP 800-50 yakni tingkatan kemampuan dasar (*beginner*), menengah (*intermediate*), dan spesialisasi (*advanced*). Fungsi dari *training path* ini adalah sebagai panduan/pedoman bagi penyelenggara program pelatihan dalam menyelenggarakan program

pelatihan dan sebagai referensi bagi SDM insiden respons yang akan mengembangkan kompetensinya dengan cara mengikuti program pelatihan. Hasil analisis dan pemetaan kompetensi Perespons Insiden Keamanan Siber dari NICE *Framework* diberikan pada Tabel 1.

Tabel 1. Pemetaan Kompetensi NICE *Framework*

Level	*Task* (T)	*Knowledge* (K)	*Skill* (S)	*Abilities* (A)
Advanced (Tinggi)	T0047, T0161, T0163, T0175, T0262, T0312, T0278, T0503	K0002, K0006, K0021, K0026, K0230, K0259	S0003, S0077, S0079, S0080, S0365	A0121, A0128
Intermediate (Menengah)	T0041, T0164, T0170, T0214, T0279	K0062, K0042, K0046, K0058, K0034, K0041, K0070, K0565, K0106, K0624, K0179	S0047, S0078, S0173	
Beginner (Dasar)	T0233, T0246, T0395, T0510	K0001, K0003, K0004, K0005, K0157, K0161, K0162, K0221, K0287, K0332		



Gambar 2 *Training Path* Perespons Insiden Keamanan Siber

Setelah menentukan klasifikasi dari kompetensi NICE *Framework*, selanjutnya adalah penyusunan pelatihan-pelatihan tersebut ke dalam bentuk *Training Path* yang ditampilkan pada Gambar 2.

4.2. Analisis Kebutuhan Diklat (AKD)

Proses AKD pada penelitian ini diawali dengan melakukan klarifikasi terhadap masalah-masalah atau keluhan yang ada pada pegawai terkait dengan terlebih dahulu menentukan responden dan metode yang digunakan dalam pengumpulan data. Jumlah responden yang berpartisipasi dalam penelitian ini sebanyak 14 responden yang merupakan Perespons Insdien Keamanan Siber pada Lembaga XYZ. Langkah selanjutnya adalah dengan memetakan kompetensi Perespons Insiden Keamanan Siber berdasarkan NICE *Framework*. Uraian tugas dan kompetensi dari Perespons Insiden Keamanan Siber sudah dijelaskan dalam NICE *Framework* yang terdiri dari deskripsi tugas terkait, kemudian pengklasifikasian dari kompetensi-kompetensi apa saja yang terkait dengan deskripsi tugas tersebut. Kompetensi tersebut terbagi ke dalam 4 jenis yaitu *tasks*/tugas, *knowledge*/pengetahuan, *skills*/keahlian, dan *abilities*/kemampuan. Masing-masing kompetensi yang sesuai dikelompokkan berdasarkan jenisnya dan ditulis dengan keterangan kode ID Kompetensi, seperti pada Tabel 2.

Tabel 2. Uraian Tugas dan kompetensi pada NICE *Framework*

Work Role Name	*Cyber Defense Incident Responder*
Work Role ID	PR-CIR-001
Speciality Area	*Incident Response* (CIR)
Category	*Protect and Defend* (PR)
Work Role Description	*Investigates, analyzes, and responsds to cyber incidents within the network environment or enclave*
Tasks	T0041, T0047, T0161, T0163, T0164, T0170, T0175, T0214, T0233, T0246, T0262, T0278, T0279, T0312, T0395, T0503, T0510
Knowledges	K0001, K0002, K0003, K0004, K0005, K0006, K0021, K0026, K0033, K0034, K0041, K0042, K0046, K0058, K0062, K0070, K0106, K0157, K0161, K0162, K0167, K0177, K0179, K0221, K0230, K0259, K0287, K0332, K0565, K0624
Skills	S0003, S0047, S0077, S0078, S0079, S0080, S0173, S0365
Abilities	A0121,A0128

Berdasarkan uraian kompetensi pada Tabel 2 kemudian dilakukan pemetaan dengan uraian tugas yang dimiliki Perespons Insiden Keamanan Siber Lembaga XYZ. Selanjutnya, dilakukan pengambilan data kepada setiap responden dengan wawancara dan kuesioner sehingga diperoleh daftar kesenjangan kompetensi dari setiap responden. Hasil akhir dari AKD ini adalah diperolehnya 10 (sepuluh) kompetensi yang menjadi mayoritas senjang dari seluruh responden.

4.3. Metode USG

Metode USG digunakan untuk mengetahui kompetensi yang menjadi prioritas untuk dilakukan pengembangan kompetensi berupa perancangan

program pelatihan. Dari 10 kesenjangan kompetensi pada tahap AKD kemudian dilakukan pengolahan menggunakan Metode USG dengan memberi nilai untuk setiap 10 komptensi berdasarkan taraf urgensinya seperti pada Tabel 3 berikut ini.

Tabel 3. Perhitungan Metode USG

No.	Kode	U (1-5)	S (1-5)	G (1-5)
1.	T0161	4,42	4,5	4,28
2.	T0163	4,28	4,28	4
3.	T0175	4	3,85	3,71
4.	T0214	3,78	3,57	3,57
5.	T0233	3,58	3,4	3,57
6.	T0279	2,78	3	2,92
7.	T0503	2,85	3	3,14
8.	S0365	2,78	2,92	3,5
9.	A0121	2,92	3,07	3,5
10.	A0128	3,57	3,21	3,42

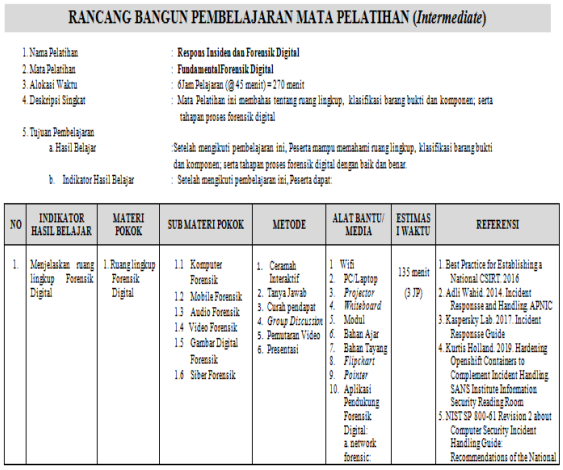
Berdasarkan hasil Metode USG, didapatkan 5 kompetensi yang menjadi prioritas karena memiliki nilai U, S, dan G yang tinggi dibanding kompetensi lainnya. Lima kompetensi ini adalah dasar dari penyusunan program pelatihan Kompetensi yang mendesak tersebut adalah 5 kompetensi dengan nilai tertinggi untuk ketiga aspek pada Metode USG baik dari *urgency*, *seriousness*, dan *growth* seperti yang tertera pada Tabel 4 berikut.

Tabel 4. Hasil Metode USG

No.	Kode	Kompetensi
1.	T0161	Melakukan analisis *file log* dari berbagai sumber (seperti *log individualhost*, lalu lintas jaringan, *log firewall*, dan *log* IDS) untuk mengidentifikasikan ancaman yang mungkin terjadi pada keamanan jaringan
2.	T0163	Mampu melakukan penanganan insiden pertahanan siber termasuk menentukan lingkup, kepentingan, dan dampak potensial, mengidentifikasi kerawanan spesifik, serta membuat rekomendasi yang memungkinkan untuk melakukan perbaikan secara cepat (*real time*).
3.	T0175	Melakukan pekerjaan penanganan insiden pertahanan siber secara *real time* (seperti melakukan pelacakan, analisis ancaman, dan dapat memulihkan suatu jaringan secara cepat) untuk mendukung tim perespons insiden keamanan siber
4.	T0214	Mampu untuk menerima dan menganalisis peringatan pada jaringan dari berbagai sumber dalam organisasi dan menentukan kemungkinan penyebab dari adanya peringatan terhadap jaringan tersebut.
5.	T0233	Mampu melacak dan mendokumentasikan insiden pertahanan siber dari deteksi awal sampai menemukan solusi di akhir.

4.4. Formulasi Program Pelatihan

Program pelatihan dirancang berdasarkan 5 kompetensi senjang yang dihasilkan dari Metode USG. Format penulisan Program Pelatihan ini menggunakan panduan yang sesuai dan bertujuan agar hasil perancangan program pelatihan memiliki komponen-komponen yang lengkap dan jelas. Program pelatihan ini ditampilkan dalam bentuk Rancang Bangun Pembelajaran Mata Pelatihan (RBPMP) berisi antara lain berupa uraian dari tiap-tiap mata pelatihan, indikator yang dicapai, materi pokok, metode, alata bantu, jumlah jam pembelajaran dan referensi materi. Salah satu contoh mata pelatihan yang terdapat pada RBPMP dapat dilihat pada Gambar 3.



RANCANG BANGUN PEMBELAJARAN MATA PELATIHAN (*Intermediate*)

1. Nama Pelatihan : **Respons Insiden dan Forensik Digital**
2. Mata Pelatihan : **FundamentalForensik Digital**
3. Alokasi Waktu : 6Jam Pelajaran (@ 45 menit) = 270 menit
4. Deskripsi Singkat : Mata Pelatihan ini membahas tentang ruang lingkup, klasifikasi barang bukti dan komponen; serta tahapan proses forensik digital
5. Tujuan Pembelajaran
   a. Hasil Belajar : Setelah mengikuti pembelajaran ini, Peserta mampu memahami ruang lingkup, klasifikasi barang bukti dan komponen; serta tahapan proses forensik digital dengan baik dan benar.
   b. Indikator Hasil Belajar : Setelah mengikuti pembelajaran ini, Peserta dapat:

NO	INDIKATOR HASIL BELAJAR	MATERI POKOK	SUB MATERI POKOK	METODE	ALAT BANTU/ MEDIA	ESTIMASI WAKTU	REFERENSI
1.	Menjelaskan ruang lingkup Forensik Digital	1. Ruang lingkup Forensik Digital	1.1 Komputer Forensik 1.2 Mobile Forensik 1.3 Audio Forensik 1.4 Video Forensik 1.5 Gambar Digital Forensik 1.6 Siber Forensik	1. Ceramah Interaktif 2. Tanya Jawab 3. Curah pendapat 4. *Group Discussion* 5. Pemutaran Video 6. Presentasi	1. Wifi 2. PC/Laptop 3. *Projector* 4. *Whiteboard* 5. Modul 6. Bahan Ajar 7. Bahan Tayang 8. *Flipchart* 9. *Pointer* 10. Aplikasi Pendukung Forensik Digital: a. network forensic:	135 menit (3 JP)	1. Best Practice for Establishing a National CSIRT. 2016 2. Adli Wahid. 2014. Incident Responsse and Handling APNIC 3. Kaspersky Lab. 2017. Incident Responsse Guide 4. Kurtis Holland. 2019. Hardening Openshift Containers to Complement Incident Handling SANS Institute Information Security Reading Room 5. NIST SP 800-61 Revision 2 about Computer Security Incident Handling Guide Recommendations of the National

Gambar 3. RBPMP

Keseluruhan mata pelatihan yang ada pada RBPMP dapat dilihat pada Tabel 5 yang menampilkan daftar mata pelatihan dan Jumlah Jam Pembelajaran (JP) masing-masing, perlu diketahui bahwa 1 JP sama dengan 45 menit.

Tabel 5. Mata Pelatihan

No.	Mata Pelatihan	JP
1.	Fundamental Forensik Digital	6
2.	Fundamental Respons dan Penanganan Insiden	9
3.	Analisis Respons Insiden pada *File Log*	13
4.	Pemeriksaan Forensik Bukti Elektronik	18
5.	Studi Lapangan Respons Insiden dan Forensik Digital	`10
Jumlah		56

Jam Pembelajaran yang ada pada RBPMP ini ditentukan berdasarkan perhitungan klasifikasi indikator yang digunakan dalam pencapaian tujuan mata pelatihan yang dimaksud. Indikator tersebut didapatkan dari pengklasifikasian pada Taksonomi Bloom. Selain pertimbangan indikator dalam Taksonomi Bloom tersebut, faktor lain seperti materi pokok, alat bantu, dan metode yang digunakan dalam

pembelajaran juga menjadi dasar penentuan banyaknya JP yang dibutuhkan.

4.5. Metode Delphi

Hasil penelitian pada tugas akhir ini dilakukan sesuai dengan proses pengambilan keputusan menggunakan Metode Delphi. Tahapannya dimulai dari pemilihan pakar/narasumber, formulasi atau penyusunan rancangan *training path* dan program pelatihan berdasarkan NICE *Framework*, lalu tahapan iterasi, dan penyimpulan hasil akhir dari *training path* dan program pelatihan yang diperoleh dari kesepakatan para pakar pada proses Delphi ini.

a. Pemilihan Narasumber
Pada tahapan ini, pemilihan narasumber dilakukan dengan mencari pihak-pihak yang sesuai dan berkompeten dalam menentukan *training path* Perespons Insiden Kamanan Siber dan Program Pelatihan Teknis Respons Insiden Forensik Digital. Narasumber yang dipilih dalam Metode Delphi pada penelitian ini merupakan narasumber yang bekerja pada bidang keamanan siber yang terdiri dari personil Lembaga XYZ dan pihak luar lembaga yang berkompeten.

b. Penyusunan Rancangan *Training Path* dan Program Pelatihan Perespons Insiden Keamanan Siber berdasarkan NICE *Framework*
penyusunan rancangan *training path* dan program pelatihan bagi Perespons Insiden Keamanan Siber berdasarkan NICE *Framework* ditujukan sebagai bahan yang diberikan kepada pakar dengan tujuan meminta koreksi, saran, masukan, dan persetujuan. Penyusunan *training path* menggunakan kompetensi Perespons Insiden Keamanan Siber pada NICE *Framework*. Sedangkan program pelatihan berasal dari hasil AKD yang menghasilkan informasi mengenai adanya kesenjangan kompetensi pada Perespons Insiden Keamanan Siber dan kompetensi yang dinilai senjang tersebut untuk selanjutnya dianalisis menggunakan Metode USG sehingga menghasilkan kompetensi yang paling mendesak untuk dibuatkan program pelatihan.

c. Iterasi 1/*Round* 1 Metode Delphi
Seluruh pakar yang menerima dokumen program pelatihan dan *training path* Perespon Insiden Keamanan Siber memberikan umpan balik berupa kritik, saran, dan masukan terhadap dokumen tersebut. Hasil ringkasan respons yang diberikan para pakar iterasi pertama terdapat pada Tabel 6.

d. Iterasi 2/*Round* 2 Metode Delphi
Setelah modifikasi pada rancangan program pelatihan pada Iterasi I selesai, kemudian berlanjut ke Metode Delphi Iterasi 2 dengan membagikan hasil modifikasi rancangan program pelatihan kepada setiap pakar untuk dikoreksi. Pada Iterasi kedua ini, para pakar akan menanggapi hasil dari iterasi sebelumnya dan kemudian menilai apakah perubahan yang terjadi sudah disetujui atau belum. Apabila belum setuju, para pakar akan memberikan saran perubahan atau saran referensi. Hasil Iterasi 2 terdapat pada Tabel 7.

Tabel 6. Hasil Iterasi 1

No	Indikator	Pakar					Ket. Pakar
		1	2	3	4	5	
1.	Pengklasifikasian *Training Path*.	V	V	X	X	V	Pakar 3 dan 4 belum setuju dengan penyebutan dalam klasifikasi pelatihan.
2.	Jenis-jenis Pelatihan dalam *Training Path*	X	X	V	V	X	Pakar 1,2, dan 5 memberikan koreksi dan saran mengenai penambahan, pemisahan, dan penggabungan pelatihan yang ada.
3.	Model Penyusunan Program Pelatihan dalam RBPMP	V	V	V	V	V	Seluruh pakar setuju.
4.	Pelatihan *File Log*	X	X	X	V	X	Pakar 1,2,3, dan 5 memberikan koreksi tentang pemberian nama pelatihan dan materi yang diajarkan.
5.	Pelatihan CSIRT	V	X	X	V	X	Pakar 2 menyarankan untuk menambah referensi materi pelatihan ini pada sumber lainnya. Sedangkan pakar 3 dan 5 belum setuju dengan nama pelatihan dan JP.
6.	Pelatihan Forensik Digital	X	V	X	V	X	Pakar 1, 3, dan 5 belum setuju dengan nama dan isi pelatihan ini.

No	Indikator	Pakar					Ket. Pakar
		1	2	3	4	5	
7.	Pelatihan Pemeriksaan Forensik Bukti Digital	X	X	X	V	X	Pakar 1, 2, 3, dan 5 belum setuju dengan nama dan isi pelatihan ini.
8.	Pelatihan Respons Insiden Komputasi Awan	V	V	V	V	X	Pakar 5 belum setuju dnegan JP dan metode yang digunakan.
9.	Pelatihan Forensik Jaringan	V	V	V	V	X	Pakar 5 belum setuju dengan JP pelatihan ini.

Ket: : Tak Setuju :Setuju

Tabel 7. Hasil Iterasi 2

No	Indikator	Pakar					Ket. Pakar
		1	2	3	4	5	
1.	Pengklasifikasian *Training Path*.	V	V	V	X	V	Pakar 3 menjadi setuju. Pakar 4 belum setuju dengan penyebutan klasifikasi tersebut.
2.	Jenis-jenis Pelatihan dalam *Training Path*	V	V	V	V	V	Pakar 1,2, dan 5 menjadi setuju.
3.	Model Penyusunan Program Pelatihan dalam RBPMP	V	V	V	V	V	Seluruh pakar setuju.
4.	Pelatihan Analisis Respons pada*File Log*	V	V	X	V	V	Pakar 1,2, dan 5 menjadi setuju. Sedangkan pakar 3 belum setuju dengan nama pelatihan.
5.	Pelatihan CSIRT	V	X	X	V	X	Pakar 2,3, dan 5 menjadi setuju.
6.	Pelathan Fundamental Forensik Digital	X	X	X	V	V	Pakar 5 menjadi setuju. Pakr 2 menjadi tidak setuju. Pakar 1 dan 3 belum setuju dengan media dan materi yang digunakan.
7.	Pelatihan Pemeriksaan Forensik Bukti Digital	V	X	V	V	V	Pakar 1,3, dan 5 menjadi setuju Pakar 2 belum setuju dengan tujuan pelatihan.
8.	Pelatihan Respons Insiden Komputasi Awan	V	V	V	V	V	Pakar 5 menjadi setuju.
9.	Pelatihan Forensik Jaringan	V	V	X	V	V	Pakar 5 menjadi setuju. Pakar 3 menjadi tidak setuju dengan alat bantu yang digunakan dlm pelatihan ini.
10.	Unsur persyaratan dan indikator dalam Program Pelatihan	V	X	V	V	X	Pakar 2 dan 5 belum setuju dengan persyaratan dan kompetensi yang harus dicapai dalam pelatihan.
11.	Pelatihan Studi Lapangan Respons Insiden Forensik Digital	V	V	X	V	V	Pakar 3 belum setuju tentang nama dan indikator pencapaian yang diharapkan.

Ket: : Tak Setuju :Setuju

e. Iterasi 3/*Round* 3 Metode Delphi

Pada Iterasi ketiga ini para pakar sudah hampir menyetujui seluruh aspek yang dimodifikasi pada rancangan program pelatihan. Beberapa hal yang masih menjadi pertimbangan akan dianalisis dengan mengambil pendapat dari mayoritas pakar yang memberikan argumen atau pendapat mengenai hal tersebut. Hasil analisis Iterasi 3 terdapat pada Tabel 8.

Tabel 8. Hasil Iterasi 3

No	Indikator	Pakar					Ket. Pakar
		1	2	3	4	5	
1.	Pengklasifikasian *Training Path*.	V	V	V	V	V	Pakar 4 menjadi setuju dengan sedikit catatan tentang kata hubung yang digunakan dalam penamaan.
2.	Jenis-jenis Pelatihan dalam *Training Path*	V	V	V	V	V	Seluruh pakar setuju.
3.	Model Penyusunan Program Pelatihan dalam RBPMP	V	V	V	V	V	Seluruh pakar setuju dengan sedikit catatan dari Pakar 3.

No	Indikator	Pakar					Ket. Pakar
		1	2	3	4	5	
4.	Pelatihan Analisis Respons pada *File Log*	V	V	V	V	V	Pakar 3 menjadi setuju dan seluruh pakar setuju
5.	Pelatihan CSIRT	V	V	X	V	V	Pakar 2 dan 5 menjadi setuju. Pakar 3 masih belum setuju.
6.	Pelathan Fundament-al Forensik Digital	V	V	V	V	V	Pakar 2 dan 5 menjadi setuju, dan seluruh pakar setuju.
7.	Pelatihan Pemeriksa-an Forensik Bukti Digital	V	V	V	V	V	Pakar 2 menjadi setuju, dan seluruh pakar setuju
8.	Pelatihan Respons Insiden Komputasi Awan	V	V	V	V	V	Seluruh pakar setuju.
9.	Pelatihan Forensik Jaringan	V	V	V	V	V	Pakar 3 menjadi setuju, dan seluruh pakar setuju
10.	Unsur persyaratan dan indikator dalam Program Pelatihan	V	V	V	V	V	Pakar 2 menjadi setuju. Pakar 5 setuju dengan sedikit catatan.
11.	Pelatihan Studi Lapangan Respons Insiden Forensik Digital	V	V	V	V	V	Seluruh pakar setuju dengan sedikit catatan dari Pakar 3.

Ket: : Tak Setuju :Setuju

f. Hasil Akhir

Pada Iterasi ketiga ini didapatkan hasil bahwa seluruh pakar sudah setuju dengan konsep dan konten yang ada pada *training path* dan program pelatihan yang dirancang. Meskipun terdapat sedikit tambahan saran dan masukan, namun pada dasarnya tetap seluruh pakar telah memberikan persetujuan atas hasil yang dirancang. Dengan selesainya Iterasi ketiga ini, maka berakhir pula Metode Delphi yang dilakukan dalam penelitian ini yang sesuai dengan pembatasan awal, yakni hingga Iterasi ketiga. Hasil terakhir modifikasi yang dilakukan pada Iterasi ketiga ini merupakan dokumen terakhir yang dijadikan sebagai hasil akhir dari penelitian ini yang berisikan *training path* dan rancangan program pelatihan bagi Perespons Insiden Keamanan Siber yang terdiri dari 5 RBPMP. Hasil akhir yang merupakan *training path* dan rancangan program pelatihan ini untuk selanjutnya disusun sebagai draf rekomendasi bagi Pusdiklat pada Lembaga XYZ.

5. SIMPULAN DAN SARAN

Dari Analisis Kebutuhan Diklat (AKD) terhadap Perespons Insiden Keamanan Siber yang ada di XYZ diketahui bahwa seluruh responden memiliki kesenjangan kompetensi terhadap NICE *Framework*. Terdapat 10 kompetensi yang menjadi mayoritas senjang dari keseluruhan responden yang dijadikan dasar utama dalam proses pengembangan kompetensi berupa perancangan program pelatihan. Rancangan program pelatihan ini dihasilkan dari rekomendasi AKD yang menghasilkan 10 kompetensi senjang. Dengan metode *Urgency*, *Seriousness*, *Growth* (USG), rangkaian proses menghasilkan Program Pelatihan Respons Insiden dan Forensik Digital bagi Perespons Insiden Keamanan Siber XYZ yang terdiri dari 5 Mata Pelatihan yang terkonsepkan dalam Rancang Bangun Pembelajaran Mata Pelatihan (RBPMP). Hasil dari *Training Path* Perespons Insiden Keamanan Siber yang dibuat menampilkan alur dari penjenjangan pelatihan yang bisa diikuti apabila menjadi seorang Perespons Insiden Keamanan Siber. *Training Path* tersebut berisi penjenjangan dari pelatihan berklasifikasi dasar/*beginner*, menengah/*intermediate*, dan ahli/*advanced* berdasarkan kompetensi pada NICE *Framework*

Penelitian ini dapat dilanjutkan dengan meneruskan tahapan dari Model ADDIE agar hasil penelitian yang dihasilkan lebih baik lagi, yakni tahap *analyze*, *design*, *development*, *implementation*, dan *evaluation* agar program pelatihan yang sudah dirancang dapat diimplementasikan dan dievaluasi mengenai materi dan pelaksanaannya. Penelitian ini juga dapat diteruskan dengan melanjutkan perancangan seluruh program pelatihan yang terdapat pada *Training Path* Perespons Insiden Keamanan Siber agar didapatkan program pelatihan yang lengkap dan menyeluruh.

Referensi:

[1] Pribadi, B. A., Desain dan Pengembangan Program Pelatihan Berbasis Kompetensi Implementasi Model ADDIE. 1st edn. Jakarta: Prenada Media Group, 2014.

[2] Mas'ud, M. I., Peta Kompetensi Deputi Bidang Penanggulangan dan Pemulihan Badan Siber dan Sandi Negara Berdasarkan NICE Framework, Jakarta, 2018.

[3] Usman F., Program Pendidikan dan Pelatihan Pegawai dalam Upaya Meningkatkan Kompetensi Sumber Daya Manusia, 2017.

[4] Sari, B. K., Desain Pembelajaran Model ADDIE dan Implementasinya dengan Teknik Jigsaw Desain Pembelajaran di Era ASEAN Economic Community (AEC) untuk Pendidikan Indonesia Berkemajuan, pp 87-102, 2017.

[5] Dick, W & Carey, L., *The Systematic Design of Instruction* (4th Ed), Harper Collins College Publishers, 1996.

[6] Newhouse, W . *et al. National Initiative for Cybersecurity Education* (NICE) *Cybersecurity Workforce* Framework, doi:10.6028/NIST.SP.800-181, 2017.

[7] Widiasih, Wiwin., dkk, Identifikasi Risiko pada Saat Implementasi *Lean Manufacturing* dengan Metode Delphi, Prosiding Studi MMT-ITS, Surabaya, 1 Agustus 2015.

[8] Bloom, B. S., *Taxonomy of Educational Objectives: The Clasification of Educational Goals, Hanbook i Cognitive Domain*, 1956.

[9] Sugiyono, Metode Penelitian Kuantitatif, Kualitatif, dan R&D, Afabeta, 2011.

[10] Nugroho, A. A., Pengembangan Perangkat Pembelajaran Matematika Berbasis SMART dengan Strategi pada Materi Segitiga Kelas VII, AKSIOMA (Vol. 2, NO 2/September), 2011Y.

Implementasi S\Key *Protocol* pada Raspberry Pi sebagai Sistem Kontrol Akses Berbasis *Client-Server* untuk Mengatasi *Replay Attack*

Alfian Andre Anto[1)] dan Dion Ogi[2)]
(1) *Badan Siber dan Sandi Negara / alfian.andre@bssn.go.id*
(2) *Politeknik Siber dan Sandi Negara / dion.ogi@poltekssn.ac.id*

Abstrak

Pada umumnya metode autentikasi yang sering digunakan untuk meningkatkan faktor keamanan pada jaringan komunikasi sistem kontrol akses berbasis client-server *adalah dengan menggunakan* password. Password *banyak digunakan karena memiliki kelebihan, yaitu bersifat mudah diingat oleh pengguna dan sederhana. Namun, teknik autentikasi* password *pada sistem kontrol akses berbasis* client-server *yang mempunyai nilai* credential *statis memiliki kekurangan yaitu rawan terhadap* Replay Attack. *Pada penelitian ini, dilakukan pembuktian eksperimental pada sistem kontrol akses berbasis* client-server *yang telah menerapkan* S\Key Protocol *dalam mengatasi* Replay Attack. Client *yang digunakan adalah Raspberry Pi sedangkan* server *yang digunakan adalah* Personal Computer *(PC). Hasil penerapan* S\Key Protocol *pada sistem kontrol akses berbasis* client-server *dapat mengatasi* Replay Attack.

Kata Kunci: *Replay Attack* (1), Sistem Kontrol Akses Berbasis *Client-Server* (2), S\Key *Protocol* (3).

1. PENDAHULUAN

Seiring dengan perkembangan zaman sistem kontrol akses dapat dibagi menjadi dua berdasarkan autentikasi yang dilakukan, yaitu kontrol akses berbasis *client* dan kontrol akses berbasis *client-server* [1]. Terdapat lima teknik autentikasi yang biasanya digunakan pada sistem kontrol akses, salah satunya adalah teknik *two factor authentication* [2]. Teknik *two factor authentication* digunakan karena dapat menambah tingkat keamanan akun pengguna dengan cara pengguna harus memasukkan dua faktor autentikasi agar sistem dapat memeriksa apakah pengguna yang diverifikasi adalah pengguna yang benar dan sah [2], [3].

Pada umumnya metode autentikasi yang sering digunakan untuk meningkatkan faktor keamanan pada jaringan komunikasi yang tidak aman adalah dengan menggunakan *password* atau PIN, tanda tangan elektronik, *smart cards*, dan *fingerprints* [4]. Metode autentikasi *password* atau PIN bayak digunakan karena metode ini paling mudah dan paling sederhana [5], [6]. Namun, menggunakan *password* atau PIN yang nilai *credential*-nya bersifat statis sebagai nilai yang diautentikasi oleh sistem berbasis *client-server* rawan terhadap serangan yang dijalankan pada jalur komunikasi seperti *replay attack* [5], [6].

Skema autentikasi *One Time Password* (OTP) dapat mengatasi kerawanan *password* atau PIN akan *replay attack* [7]. OTP tahan akan *replay attack* karena sifat OTP yang hanya sekali pakai. Salah satu jenis OTP yang dapat digunakan untuk mengatasi *replay attack* adalah S\Key *Protocol* [6], [7]. S\Key *Protocol* membutuhkan fungsi *hash* untuk membangkitkan nilai OTP [6]. Pada awalnya fungsi *hash* yang digunakan pada sistem S\Key *Protocol* adalah MD4, namun dengan seiring berkembangnya zaman telah ditemukan banyak kelemahan dari MD4, oleh karena itu maka tidak disarankan untuk menggunakan MD4. Fungsi *hash* yang memiliki ketahanan pada *Key Length Extension Attack*, *Collision Attack*, *Preimage Attack*, dan *Second Preimage Attack* yang merupakan *Secure Hash Standard* dari *National Institute of Security and Technology* (NIST) adalah SHA-3 [8].

Berdasarkan latar belakang permasalahan tersebut, maka penulis melakukan penelitian mengenai pengaruh implementasi dari S\Key *Protocol* di Raspberry Pi sebagai sistem kontrol akses berbasis *client-server* terhadap *replay attack*. Input *password* dari pengguna akan dijadikan sebagai *seed* masukan S\Key *Protocol* untuk membangkitkan nilai *one time password*. Fungsi *hash* yang akan digunakan pada S\Key *Protocol* adalah SHA-3. Sistem kontrol akses ini juga akan menerapkan *two factor authentication* berupa masukan kartu RFID dan *password*. Sistem kontrol akses ini dibangun dengan menggunakan Raspberry Pi sebagai sisi *client*, sedangkan pada sisi *server* menggunakan *Personal Computer* (PC) yang berfungsi untuk mengorganisir *database* akun pengguna. Raspberry Pi dan PC dihubungkan dengan jaringan nirkabel lokal dengan protokol TCP/IP.

2. LANDASAN TEORI

2.1. Sistem Kontrol Akses Berbasis *Client-Server*

Kontrol akses atau kendali akses merupakan sistem yang digunakan untuk komunikasi antara pengguna dan sistem dengan sistem lain serta sumber

daya yang ada di dalamnya [9]. Menurut *Certified Information Sistem Security Profesional* (CISSP), kontrol akses merupakan suatu proses untuk mengatur atau mengendalikan siapa saja yang berhak untuk mengakses suatu sumber daya tertentu yang ada pada suatu sistem. Tujuan dari kontrol akses sendiri adalah memastikan sistem hanya dapat diakses oleh entitas yang benar saja [9].

Kontrol akses berdasarkan autentikasi dapat dibagi menjadi dua, yaitu kontrol akses berbasis *client* dan kontrol akses berbasis *client-server*. Sistem kontrol akses berbasis *client-server* adalah salah satu contoh sistem kontrol akses yang memanfaatkan jaringan baik nirkabel ataupun kabel untuk melakukan autentikasi kepada pengguna [1]. Sistem kontrol akses berbasis *client-server* mempunyai kelebihan dibandingkan dengan sistem kontrol akses berbasis *client* yaitu sistem dapat mengelola data akun pengguna secara terpusat di satu komputer *server*. Tujuan dari pengelolaan akun secara terpusat di komputer *server* adalah untuk mengurangi biaya, meningkatkan keamanan, memenuhi kepatuhan pegawai terhadap peraturan, dan meningkatkan tingkat layanan organisasi[9].

2.2. S\Key *Protocol*

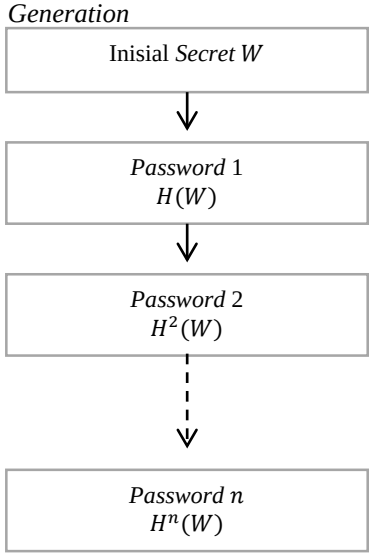
Protokol OTP S\Key khusus didesain sebagai solusi untuk pengamanan jalur komunikasi antara dua *device* yang dihubungkan dengan jalur yang tidak aman dan rawan akan penyadapan. Sistem S\Key memiliki beberapa kelebihan dibandingkan dengan sistem *one-time* dan *multi-use authentication* [10]. *Secret password* milik pengguna pada sistem *one time password* (OTP) S\Key tidak pernah dimasukkan ke jaringan pada saat proses *login* [10]. *Secret password* milik pengguna akan diproses terlebih dahulu oleh *client* sebelum hasil dari proses akan dikirimkan melalui jaringan kepada *server*. Proses yang harus dilewati adalah fungsi *hash* yang aman untuk menghasilkan *one time password* (OTP).

Sistem OTP S\Key terdiri dari dua tahapan, yaitu tahapan *generation* dan *authentication*. Tahapan *generation* merupakan mekanisme untuk membangkitkan *password* S\Key yang digambarkan pada Gambar 1. Tahapan *authentication* merupakan mekanisme yang dilakukan *server* untuk melakukan verifikasi OTP yang dikirim oleh pengguna, tahapan ini digambarkan pada Gambar 2. Sistem OTP S\Key menghasilkan *password* dengan cara membagi dua *output* yang dihasilkan oleh fungsi *hash* kemudian kedua bagian tersebut dilakukan *folding* dengan cara *exclusive*-OR (XOR).

Berikut adalah proses dari tahapan *password generation* dari S\Key Protocol:

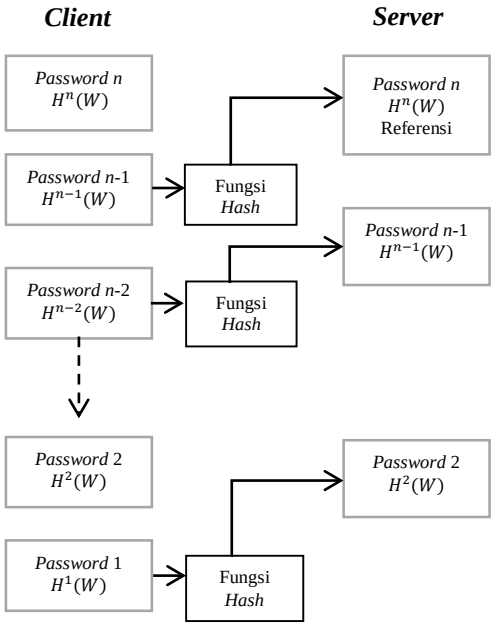
1) Kunci W dipilih oleh pengguna atau kunci W dapat dibangkitkan sendiri oleh sistem secara acak. Kunci W bersifat rahasia, apabila kunci W ini diketahui oleh pihak yang tidak berwenang maka skema OTP S\Key ini tidak berfungsi lagi.
2) Fungsi *hash* kriptografi H diterapkan sebanyak n kali terhadap kunci W, sehingga menghasilkan rantai *hash* dari dari n *one-time password*. $H(W)$, $H^2(W)$,..., $H^n(W)$. Proses pembangkitan *one-time password* ini dilakukan di sisi *client*. Jumlah n yang akan digunakan pada penelitian ini adalah sebanyak 100.
3) Client mempunyai sejumlah n *password* yang diurutkan terbalik : $H^n(W)$, $H^{n-1}(W)$, …, $H^{n-2}(W)$, $H(W)$.
4) Password $H^n(W)$ dikirimkan kepada *server* melalui jaringan dan kemudian *password* tersebut disimpan di *database server*.

a) *Generation*



Gambar 1 Fase *generation* S\Key *Protocol*

b) *Authentication*



Gambar 2 Fase *authentication* S\Key *Protocol*

Berikut adalah penjelasan dari fase *authentication* S\Key *Protocol* :

1) Pengguna mempunyai *password* $H^{n-1}(W)$ yang terdapat dalam daftar urutan *password*. Kemudian $H^{n-1}(W)$ dikirimkan ke *server* melalui jaringan komunikasi yang tidak aman.
2) *Server* menghitung nilai dari $H(H^{n-1}(W))$, $H^{n-1}(W)$ adalah *password* yang diterima dari sisi *client*. Jika $H(H^{n-1}(W))$ menghasilkan *password* yang sama dengan $H^{n}(W)$ yang disimpan di *server*, maka autentikasi berhasil. Kemudian *server* menghapus $H^{n}(W)$ dari *database* dan menggantikannya dengan $H^{n-1}(W)$ sebagai referensi.
3) Proses yang sama dilakukan berulang kali sampai dengan pembangkitan *password* yang dilakukan oleh sisi *client* tersisa sebanyak satu kali.

2.3. *Replay Attack*

Replay attack adalah salah satu bentuk serangan yang dilakukan oleh penyerang dengan cara menangkap suatu data yang ada pada protokol komunikasi pada waktu ke *t*, kemudian data tersebut dikirimkan oleh penyerang kepada pihak tujuan pada waktu ke *t+1* [6]. Tujuan dari serangan *replay attack* adalah pengguna tidak sah ingin berpura-pura menjadi pengguna yang sah [2]. Berikut adalah penjelasan dari langkah-langkah untuk melakukan *replay attack*:

a) *Client* mengirim paket data ke *server* melalui jaringan nirkabel pada waktu ke-*t*.
b) Penyerang menangkap komunikasi yang sah antara entitas yang sedang berkomunikasi pada waktu ke-*t*.
c) Penyerang mengirimkan ulang paket data yang diterima ke salah satu entitas dengan tujuan untuk berpura-pura menjadi pengguna yang sah pada waktu ke- *t+1*.

3. METODOLOGI PENELITIAN

Metodologi penelitian yang akan digunakan adalah metode eksperimen [11], dan menggunakan metode perancangan aplikasi atau *Sistem Development Life Cycle* (SDLC) dengan pendekatan *Waterfall Development* [12]. Metode eksperimen digunakan untuk menganalisis hasil pengujian skema S\Key *Protocol* yang diimplementasikan pada Raspberry Pi sebagai kontrol akses berbasis *client-server* dalam mengatasi *replay attack*. Metode rancang bangun *waterfall development* pada digunakan sebagai panduan membangun sistem dengan baik dan benar. Terdapat dua produk yang akan dihasilkan pada penelitian ini yaitu kelompok eksperimen dan kelompok kontrol. Kelompok eksperimen adalah sistem kontrol akses yang telah menerapkan S\Key *Protocol*. Kelompok kontrol adalah sistem kontrol akses yang tidak menerapkan S\Key.

Terdapat 7 tahapan pada metode penelitian eksperimen yang digunakan, yaitu *Select A Topic, Identity The Research, Conduct A Literature Search, State The Research Question (Hypothesis*), *Determine The Research Design*, *Determine Methods*, dan *Determine Data Analysis Techniques*.

Untuk memudahkan pemahaman maka dilakukan pengelompokan tujuh fase menjadi tiga tahapan, yaitu tahapan perencanaan mencakup *Select A Topic*, *Identity The Research Problem*, *Conduct A Literature Search*, *State The Research Questions (or Hypothesis*), dan *Determine the Research Design*. Tahapan persiapan hanya mencakup *Determine Methods*. Pada tahapan pelaksanaan hanya mencakup *Determine Data Analysis Techniques*. Pada Tabel 1 diberikan ringkasan dari metodologi penelitian eksperimen yang dilakukan pada penelitian ini.

Tabel 1 Ringkasan metodologi penelitian eksperimen

Fase Penelitian		Metode	Output
Perencanaan	*Select a Topic*	Studi literatur	Autentikasi pada sistem kontrol akses berbasis *client- server*
	Identity The Research Problem	Studi literatur	*Password* bersifat statis pada kontrol akses berbasis *client-server* rawan akan *Replay Attack*
	Conduct a Literatur Search	Studi literatur	S\Key *Protocol*
	State The Research Question (*or Hypo-thesis*)	• Diskusi • Konsultasi • Studi literatur	Apakah implementasi S\Key *Protocol* pada kontrol akses berbasis *client-server* dapat mengatasi *Replay Attack*?
	Determine The Research Design	Studi Literatur	*Quasi Experimental Design*
Persiapan	*Determine Methods*	Membangun sistem dengan mengguna-kan SDLC *Waterfall* Development	Sistem kontrol akses kelompok kontrol dan kelompok eksperimen
		Merancang skenario *Replay Attack*	Skenario *Replay Attack*
Pelaksanaan	*Determine Data Analysis Technique*	• *Replay Attack* pada kelompok eksperimen • *Replay Attack* pada kelompok kontrol • Komparasi hasil *Replay Attack* pada kelompok kontrol dan kelompok eksperimen	Persentase keberhasilan kelompok eksperimen dibandingkan dengan kelompok kontrol dalam mengatasi *Replay Attack*

4. PEMBANGUNAN SISTEM

4.1. Perencanaan (*Planning*)

Tahapan perencanaan yang dilakukan merupakan tahapan perencanaan untuk membangun sistem kontrol akses yang telah menerapkan S\Key *Protocol* (kelompok eksperimen). Tahap perencanaan terdiri dari dua bagian, yaitu Inisiasi Proyek dan Manajemen Proyek. Pada tahapan inisiasi proyek, peneliti menentukan bahasa pemrograman yang digunakan untuk membangun sistem. Sistem dibangun menggunakan bahasa pemrograman *python* 3.5.1, karena *python* merupakan bahasa yang banyak digunakan untuk *Network Programming*. *Network Programming* digunakan agar *client* dan *server* dapat terhubung pada suatu jaringan dan dapat saling bertukar informasi. Pada tahapan manajemen proyek, peneliti membuat jadwal pengerjaan sistem, agar sistem dapat dibangun secara terstruktur.

4.2. Analisis (*Analysis*)

Tahapan analisis merupakan tahapan untuk menentukan kebutuhan fungsional dan non-fungsional dari kelompok eksperimen yang akan dibangun. Pada tahapan ini terdiri dari dua bagian, yaitu identifikasi kebutuhan sistem dan penentuan kebutuhan sistem.

a) Identifikasi Kebutuhan

Identifikasi kebutuhan dilakukan pada literatur utama dan *benchmarking* pada penelitian sebelumnya. Hasil identifikasi kebutuhan dari literatur utama S\Key *Protocol* adalah sebagai berikut:

1) Sistem menyediakan proses pendaftaran
2) Sistem menyediakan proses *login*
3) Sistem menyediakan proses inisialisasi *password*
4) Sistem dapat mengacak posisi karakter yang ditampilkan setiap sesi *login*
5) Sistem menggunakan nilai OTP untuk autentikasi
6) Sistem menghasilkan nilai OTP dari operasi fungsi *hash*

Proses *benchmarking* dilakukan sistem kontrol akses pada penelitian sebelumnya yaitu milik taufiq [13], hidayat [14], fauzi [15], dan habibi [16]. Proses *benchmarking* menghasilkan fitur-fitur umum yang ada di sistem kontrol akses. Berikut adalah fitur-fitur umum yang terdapat pada sistem kontrol akses:

1) Sistem menyediakan proses *login*
2) Sistem menyediakan proses pendaftaran akun
3) Sistem menyediakan proses ganti *password* akun.
4) Sistem dapat menerima akses dari pengguna yang benar.
5) Sistem dapat menampilkan notifikasi keberhasilan dari setiap proses yang dilakukan oleh pengguna.

b) Penentuan Kebutuhan Sistem

Berdasarkan identifikasi kebutuhan pada sub bab diatas, dihasilkan kebutuhan fungsional dan kebutuhan non-fungsional . Tabel 2 adalah tabel kebutuhan fungsional sistem. Tabel 3 adalah tabel kebutuhan non-fungsional sistem. Tabel 2 berisi kebutuhan inti yang harus ada pada sistem, sedangkan Tabel 3 berisi tentang kebutuhan pendukung yang ada pada sistem.

Tabel 2 Kebutuhan fungsional

Kebutuhan Fungsional	Kelompok Eksperimen	Kelompok Kontrol
Sistem menyediakan layanan pendaftaran	V	V
Sistem menyediakan layanan *login*/autentikasi	V	V
Sistem menyediakan layanan mengganti *password*	V	V
Sistem menggunakan nilai OTP untuk autentikasi	V	-
Sistem dapat menghasilkan nilai OTP dari operasi fungsi *hash*	V	-

Tabel 3 Kebutuhan non-fungsional sistem

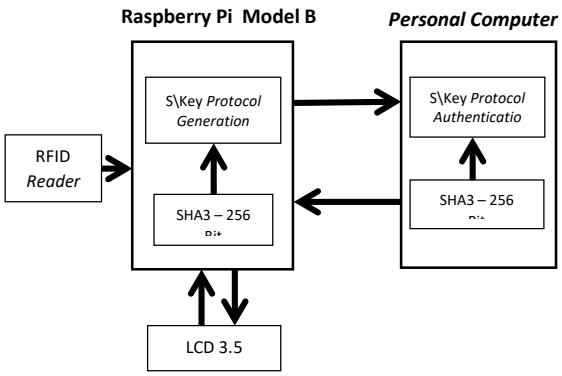
Kebutuhan Non-Fungsional	Kelompok Eksperimen	Kelompok Kontrol
OTP dibangkitkan dengan menggunakan fungsi *hash* SHA-3	V	-
Sistem menerima *input two factor authentication* pengguna	V	V
Raspberry Pi 3 model B digunakan sebagai *client* dan *Personal Computer* (PC) sebagai *server*	V	V

4.3. Perancangan (*Design*)

Tahapan perancangan merupakan tahapan yang bertujuan untuk menggambarkan sistem kontrol akses kelompok eksperimen yang akan dibangun kedalam bentuk grafis. Pada tahapan ini terdiri dua bagian, yaitu gambaran umum sistem dan rincian proses sistem kontrol akses.

a) Gambaran Umum Sistem

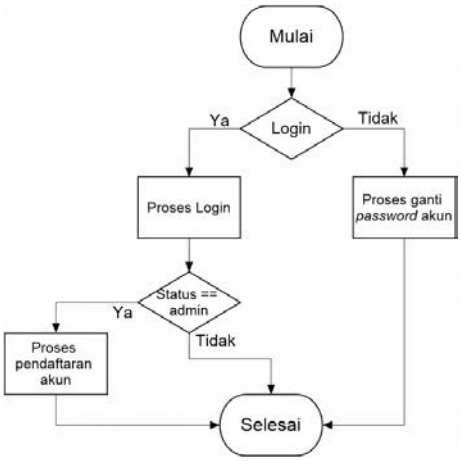
Penggambaran umum sistem kontrol akses yang akan dibangun menggunakan *block diagram* agar mudah untuk dipahami. Secara umum sistem kontrol akses mempunyai empat layanan utama, yaitu *login*, pendaftaran akun, ganti *password* akun, dan inisialisasi *password*. Gambar 3 adalah gambaran umum dari sistem kontrol akses yang akan dibangun, Raspberry Pi digunakan sebagai *server* ditambahkan komponen yaitu servo, LCD 3.5 Inch, dan RFID USB *Reader*, *Personal Computer* digunakan sebagai *server*.



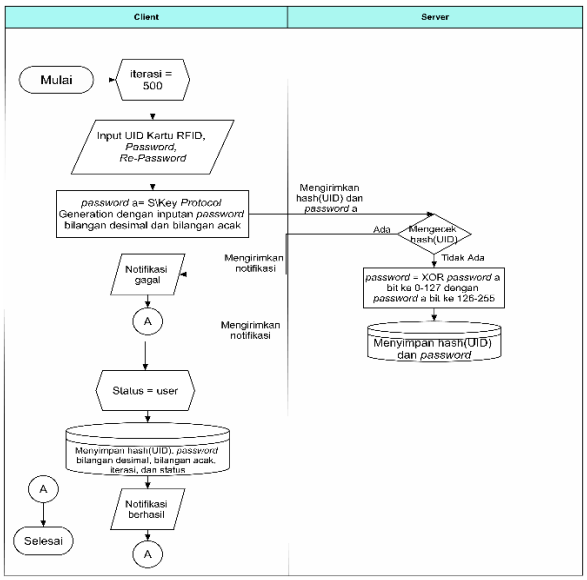
Gambar 3 Gambaran umum sistem

b) Rincian proses *system control* akses

Berdasarkan gambaran umum sistem kontrol akses pada Gambar 3, proses keseluruhan sistem kontrol akses mempunyai empat layanan, yaitu *login*, pendaftaran akun, ganti *password* akun, dan inisialisasi *password*.

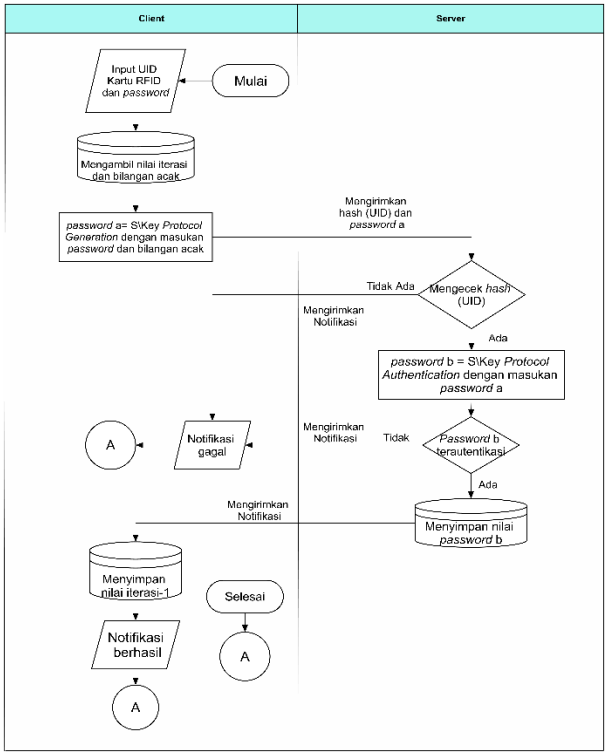


Gambar 4 Proses keseluruhan sistem

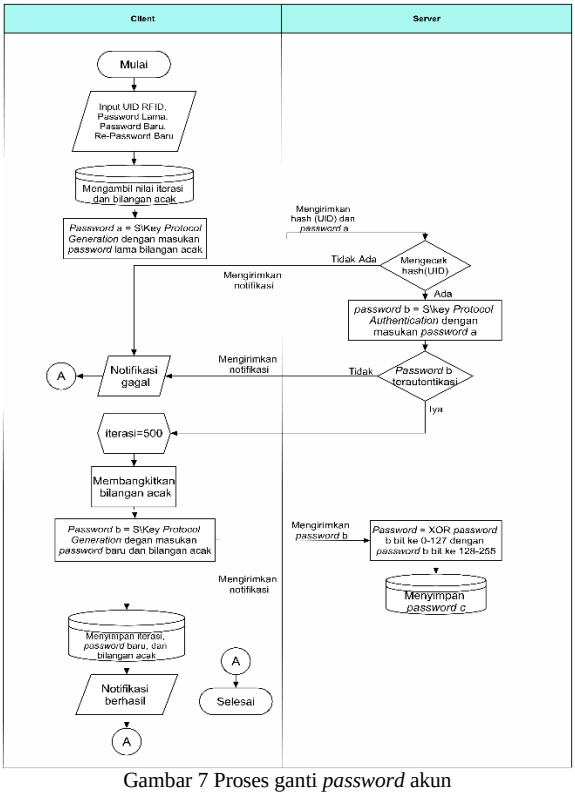


Gambar 5 Proses pendaftaran akun

Pada sub-bab ini dijelaskan setiap layanan yang terdapat pada sistem kontrol akses dan proses keseluruhan yang terdapat pada sistem kontrol akses dengan bantuan *Low Level Flowchart*. Gambar 5 adalah gambar proses pendaftaran akun, *admin* melakukan mendaftarkan akun yang belum pernah didaftarkan ke sistem.



Gambar 6 Proses *login*



Gambar 7 Proses ganti *password* akun

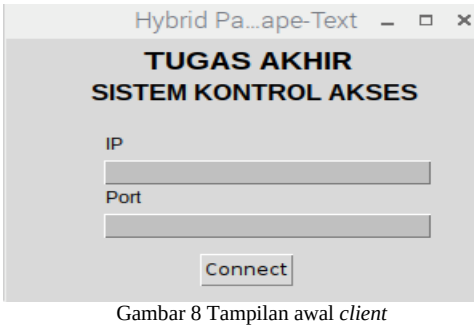
Gambar 6 adalah gambar proses *login*, pengguna atau *admin* melakukan autentikasi agar dapat masuk ke sistem. Gambar 7 adalah gambar proses ganti *password* akun, pengguna atau *admin* melakukan ganti *password* pada akun miliknya sendiri.

4.4. Implementasi (*Implementation*)

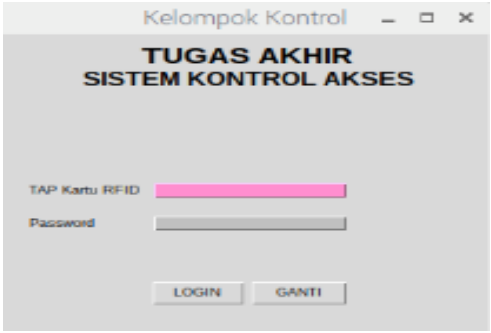
Tahapan implementasi merupakan tahapan yang menjelaskan tentang implementasi S\Key *Protocol* pada sistem kontrol akses kelompok eksperimen. Tahapan implementasi dapat dibagi menjadi dua, yaitu implementasi sistem dan pengujian sistem.

a) Implementasi sistem

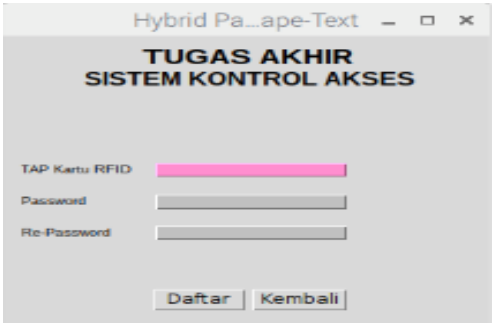
Pada bagian implementasi sistem akan dijelaskan mengenai hasil implementasi dari sistem sesuai dengan kebutuhan fungsional, kebutuhan non-fungsional, dan perancangan yang telah dilakukan pada tahap sebelumnya. Gambar 8 adalah tampilan *client* pada saat pertama kali dijalankan. Gambar 9 adalah tampilan *client* pada saat proses *login*. Gambar 10 adalah tampilan *client* pada saat proses pendaftaran akun. Gambar 11 adalah tampilan *client* pada saat proses ganti *password* akun.



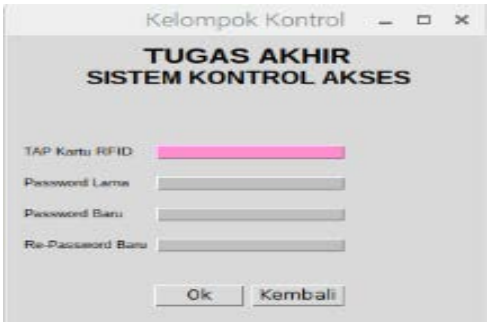
Gambar 8 Tampilan awal *client*



Gambar 9 Tampilan *login client*



Gambar 10 Tampilan pendaftaran akun *client*



Gambar 11 Tampilan ganti *password* akun *client*

b) Pengujian Sistem

Pada bagian pengujian sistem akan dijelaskan mengenai hasil pengujian sistem. Pengujian sistem ini dilakukan dengan tujuan untuk menguji sistem secara keseluruhan dan memastikan sistem dapat berjalan dengan baik. Pengujian sistem akan dilakukan pada *client* dan *server*. Pengujian sistem yang dilakukan terdiri dari tiga pengujian, yaitu *unit testing*, *integration testing*, dan *sistem testing*. *Unit testing* yang dilakukan adalah *unit testing* dengan pendekatan *black box testing*. *Integration testing* yang dilakukan pada penelitian ini adalah *user interface testing* dan *use scenario testing*.

Tabel 4 Hasil ringkasan *unit testing*

No.	Nama Fungsi	Keterangan
1.	hash()	Sesuai
2.	xor()	Sesuai
4.	login()`	Sesuai
5.	daftar()	Sesuai
6.	ganti_password()	Sesuai
7.	inisialisasi_password()	Sesuai
8.	connection()	Sesuai
9.	koneksi()	Sesuai
10.	terima()	Sesuai
11.	koneksi_ganti_password_1()	Sesuai
12.	koneksi_ganti_password_2()	Sesuai
13.	koneksi_inisialisasi_password()	Sesuai
14.	connect_ke()	Sesuai
15.	kirim()	Sesuai
16.	terima_koneksi()	Sesuai

Tabel 4 adalah hasil ringkasan dari pengujian *unit testing* yaitu mengecek kesesuaian dari fungsi didalam sistem yang telah dibangun. **Error! Not a valid bookmark self-reference.** adalah hasil ringkasan dari pengujian *user interface testing* yaitu mengecek kesesuain tampilan dari sistem. Tabel 6 adalah hasil ringkasan dari pengujian *use scenario testing* yaitu mengecek kesesuaian *scenario* yang ada pada sistem.

Tabel 5 Hasil ringkasan *user interface testing*

No	Halaman Antarmuka	Keterangan
1.	Halaman *login*	Sesuai
2.	Halaman pendaftaran akun	Sesuai
3.	Halaman ganti *password* akun	Sesuai

Tabel 6 Hasil ringkasan *use scenario testing*

No.	Nama Skenario	Keterangan
1.	Melakukan pendaftaran akun	Sesuai
2.	Melakukan *login*	Sesuai
3.	Melakukan ganti *password* akun	Sesuai

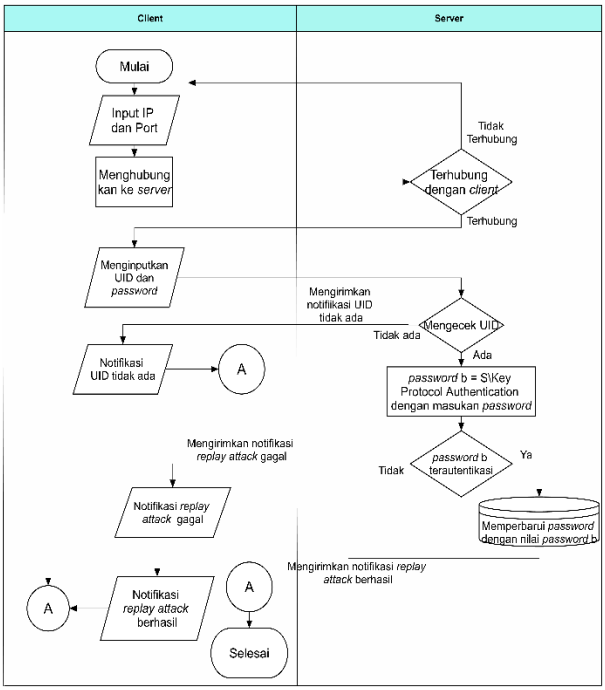
5. PEMBUKTIAN SKEMA PADA SISTEM

5.1. *Replay Attack*

Pada bagian ini penulis menjelaskan pengujian *replay attack* yang dilakukan pada sistem kontrol akses kelompok kontrol dan sistem kontrol akses kelompok eksperimen. Setelah dilakukan pengujian *replay attack*, dilanjutkan dengan analisis dari hasil *replay attack* pada sistem kontrol akses kelompok kontrol dan sistem kontrol akses kelompok eksperimen. Pengujian dilakukan dengan tiga tahapan, yaitu:

a) *Client* mengirim paket data ke server melalui jaringan nirkabel pada waktu ke-t.
b) Penyerang dengan menggunakan *tools* Wireshark menangkap paket data yang diterima oleh *server* pada poin a.
c) Penyerang dengan menggunakan aplikasi khusus mengirimkan ulang paket data yang didapat dari poin b ke *server* pada waktu ke t+1, dengan tujuan untuk berpura-pura menjadi *client* yang sebenarnya.

Aplikasi khusus yang digunakan oleh penyerang dibangun dengan menggunakan bahasa pemrograman *python* 3.5.1. Gambar 12 adalah gambaran umum dari aplikasi khusus penyerang.



Gambar 12 Gambaran umum aplikasi khusus penyerang

Pengujian dilakukan sebanyak masing-masing 20 kali pada sistem kontrol akses kelompok kontrol dan sistem akses kontrol kelompok eksperimen. Hasil dari pengujian *replay attack*, *replay attack* berhasil sebanyak 20 kali pada sistem kontrol akses kelompok kontrol dan *replay attack* gagal sebanyak 20 kali pada sistem kontrol akses kelompok eksperimen. Penyerang selalu berhasil melakukan *replay attack* pada kontrol akses kelompok kontrol dikarenakan nilai *credential* yang dikirimkan *client* ke *server* selalu tetap, sedangkan penyerang selalu gagal melakukan *replay attack* pada kontrol akses kelompok eksperimen dikarenakan nilai *credential* yang dikirimkan *client* ke *server* selalu berubah.

6. SIMPULAN DAN SARAN

Berdasarkan hasil pengujian dan analisis yang telah dilakukan dalam proses penelitian ini, maka diperoleh kesimpulan bahwa implementasi S*Key Protocol* pada sistem kontrol akses dapat mengatasi kerawanan *Replay Attack*. Percobaan *Replay Attack* sebanyak 20 kali percobaan pada sistem kontrol akses yang tidak menerapkan S\Key *Protocol* selalu berhasil, sedangkan percobaan *Replay Attack* pada sistem kontrol akses yang menerapkan S\Key *Protocol* selalu gagal.

Berdasarkan proses penelitian yang telah dilakukan, masih ditemukan kekurangan yang perlu diperbaiki. Berikut adalah saran untuk penelitian selanjutnya:

a) Membangun sistem kontrol akses dengan penambahan layanan sampai dengan layanan *authorization* dan *accountability*.
b) Sistem kontrol akses dapat dikembangkan dengan melakukan penggunaan jaringan internet untuk menghubungkan *client* dan *server*.

Referensi

[1] B. A. Forouzan, *Data Communications AND Networking*. 2013.

[2] L. C. K. Dong, *Cryptographic Protocol Security Analysis Based on Trusted Freshness*. 2011.

[3] A. J. Menezes, P. C. Van Oorschot, dan S. A. Vanstone, “APPLIED CRYPTOGRAPHY,” 1997.

[4] L. Gong, J. Pan, B. Liu, dan S. Zhao, “Journal of Computer and Sistem Sciences A novel one-time password mutual authentication scheme on sharing renewed finite random sub-passwords,” *J. Comput. Syst. Sci.*, vol. 79, no. 1, hal. 122–130, 2013.

[5] I. Liao, C. Lee, dan M. Hwang, “over insecure networks ☆,” vol. 72, hal. 727–740, 2006.

[6] Haller, “The S/Key One-Time Password Sistem,” *Netw. Work. Gr.*, hal. 1–12, 1995.

[7] D. Worth, “COK: Cryptographic one-time knocking,” *Proc. Black Hat Conf.*, hal. 19–25, 2004.

[8] M. Dworkin, “SHA-3 Standard : Permutatuin-based Hash and Extendable-Output Functions,” 2015.

[9] S. Harris, *ALL IN ONE CISSP*. 2013.

[10] N. M. Haller, “The S / Key Tm One-Time Password Sistem,” *Proc. Symp. Netw. Distrib. Syst. Secur.*, hal. 151–157, 1994.

[11] S. M. Ross dan G. R. Morrison, “EXPERIMENTAL RESEARCH METHODS,” no. January 2003, 2003.

[12] A. Dennis, B. H. Wixom, dan R. M. Roth, *Sistem Analysis and Design Fifth Edition*, vol. 5. 2012.

[13] M. Taufiq, “Implementasi One-Time Password Mutual Authentication Scheme on Sharing Renewed Finite Random Sub-Passwords pada Kontrol Akses Berbasis Client-Server untuk Mengatasi Dictionary Attack dan Replay Attack,” 2018.

[14] T. R. Hidayat, “Penerapan Skema Online Secret Sharing (OSS) Cratoon dan Algoritma Kunci Publik RSA Untuk Pengamanan Password pada Access Kontrol Ruangan,” 2017.

[15] M. Fauzi, “Rancang Bangun Prototype Kendali Akses dengan Menerapkan Skema Color Combo pada Arduino,” 2017.

[16] Habibi, "Rancang Bangun Application Locker dengan Menerapkan Skema Hybrid Password Authentication Scheme Based on Shape-Text Sebagai Solusi Alternatif Kontrol Akses Keamanan Aplikasi Pada Smartphone Android," 2017.

Rancang Bangun Prototipe Deteksi Serangan *Evil Twin* pada Jaringan Nirkabel

Fathia Mustika[1)] dan Muhammad Yusuf Bambang Setiadji[2)]
(1)Politeknik Siber dan Sandi Negara / fathia.mustika@student.poltekssn.ac.id
(2)Politeknik Siber dan Sandi Negara / muhammad.yusuf@poltekssn.ac.id

Abstrak

Teknologi wireless fidelity *(WiFi), selain memberikan manfaat, juga menimbulkan ancaman keamanan bagi penggunanya. Salah satu ancaman tersebut adalah* evil twin attack. Evil twin attack *dilakukan dengan membuat* access point *palsu yang seolah-olah memiliki SSID dan BSSID yang sama dengan* access point *asli yang berasosiasi dengan perangkat korban. Penelitian-penelitian sebelumnya telah membahas mengenai cara mendeteksi* evil twin attack, *di antaranya dengan menggunakan* white list MAC address *dan penggunaan* anomaly-based *dan* signature-based intrusion detection system *(IDS). Penggunaan IDS tersebut tidak efisien karena lebih banyak mendeteksi* false positive. *Pada penelitian ini dilakukan implementasi sistem deteksi dengan menggunakan skema Agarwal et al. untuk mendeteksi* evil twin attack. *Sistem deteksi dibangun menggunakan metodologi* System Development Life Cycle *dengan pemodelan* prototyping. *Selanjutnya, hasil implementasi diuji guna mengetahui kesesuaian fungsi sistem deteksi dengan pengujian sistem dan pengujian terhadap serangan. Berdasarkan pengujian yang dilakukan, sistem deteksi terbukti berfungsi sesuai dengan kebutuhan dan dapat mendeteksi* evil twin attack *dari keseluruhan serangan yang dilakukan.*

Kata Kunci: Intrusion detection system *(1),* Evil twin attack *(2), Skema Agarwal et al. (3)*

1. PENDAHULUAN

Jaringan WiFi yang tidak aman merupakan sarana yang sangat strategis bagi para pihak yang ingin memanfaatkan ketidakamanan tersebut. Serangan yang umum terjadi biasanya berupa *jamming* [1], *war driving*, *man-in-the-middle* [2], *pollution* [3], *authentication flood*, *de-authentication flood*, MAC *spoofing*, dan sebagainya [4]. Salah satu yang mendasari banyaknya serangan pada jaringan WiFi adalah kelemahan pada protokol 802.11. Fakta menyebutkan bahwa WiFi akan menjadi pondasi pada beberapa teknologi seperti *Internet of Things* dan *mobile cloud computing*, membuat teknologi-teknologi tersebut rentan terhadap beberapa serangan keamanan [5].

Salah satu serangan yang dapat dilakukan adalah *rogue access point* (RAP). RAP dapat dengan mudah di-*install* tanpa perlu adanya otorisasi dari pengelola jaringan [6]. RAP didesain oleh penyerang sedemikian sehingga penyerang dapat mengarahkan klien untuk masuk ke portal *log in* palsu, dan kemudian mencari data penting seperti password dan informasi kartu kredit, yang dilakukan dengan menyadap, *man-in-the-middle attack*, dan lain sebagainya.

RAP dapat dibagi menjadi empat tipe, salah satunya adalah *evil twin access point* (ETAP) [7]. Serangan dilakukan dengan mengecoh klien untuk mengasosiasikan perangkatnya pada suatu jaringan WiFi yang kemudian akan diarahkan pada suatu laman web yang telah disiapkan oleh penyerang guna mengambil informasi-informasi vital klien. Penyerang melakukan *spoofing* pada *media access control (*MAC) *address* dan juga *service set identifier* (SSID) dari *access point* (AP) yang asli untuk membuat AP *evil twin* guna mempersiapkan *evil twin attack*. Hal tersebut membuat klien hanya akan melihat jaringan WiFi tunggal. Terdeteksinya WiFi tunggal disebabkan oleh konfigurasi sistem operasi yang hanya mendeteksi AP dengan kekuatan sinyal yang lebih tinggi apabila ada beberapa AP dengan SSID sama. Apabila sinyal AP *evil twin* lebih kuat daripada AP asli, maka perangkat klien hanya akan mendeteksi AP *evil twin*. Klien kemudian akan mengasosiasikan perangkatnya pada AP *evil twin*, dengan asumsi bahwa AP tersebut memiliki koneksi internet.

Solusi yang telah ada untuk mendeteksi ETAP berupa instalasi perangkat keras tambahan, modifikasi protokol [8], mengukur karakteristik *frame* [9]–[13], dan lain-lain. Meskipun demikian, metode-metode ini kurang efektif karena dari segi biaya terlalu mahal dan masih memerlukan modifikasi protokol. Hal tersebutlah yang menjadi latar belakang penulis untuk menggunakan metodologi lain guna mendeteksi adanya ETAP.

Pada penelitian ini dibangun sebuah prototipe sistem deteksi untuk mendeteksi ETAP pada jaringan nirkabel. Prototipe sistem deteksi dibangun sebagai terobosan baru karena kondisi jaringan pada saat terjadinya *evil twin attack* hampir mirip, sehingga penerapan *signature-based* IDS dan *anomaly-based* IDS tidak dapat digunakan karena tidak akan melakukan deteksi dengan tepat [14]. Prototipe sistem deteksi tersebut diasumsikan memiliki empat komponen, yakni WiFi *sniffer*, *frame filtering*, *frame characteristics and deauthentication detector*, dan *evil twin detection* dengan harapan dapat mendeteksi *evil twin attack* secara optimal.

Pembahasan hasil penelitian ini disusun menjadi lima bagian. Pada bagian pertama dibahas pendahuluan. Bagian kedua akan dibahas mengenai teori-teori terkait penelitian. Metodologi penelitian dibahas pada bagian ketiga untuk menjelaskan langkah-langkah yang diambil dalam melakukan penelitian. Bagian inti dari penelitian ini yaitu rancang bangun sistem deteksi untuk mendeteksi *evil twin attack* pada jaringan nirkabel dibahas pada bagian keempat. Kesimpulan dan saran untuk penelitian selanjutnya dijelaskan pada bagian lima dan bagian terakhir berisi daftar pustaka yang menjadi rujukan dari penelitian.

2. KAJIAN PUSTAKA

2.1. *Intrusion Detection System*

Intrusion Detection System (IDS) merupakan sebuah layanan keamanan yang melakukan pengawasan dan menganalisis keseluruhan sistem guna menemukan suatu ketidaknormalan dengan menyediakan peringatan baik secara *real-time* ataupun tidak [15].

Pada dasarnya IDS dibangun berdasarkan *behavior* (tingkah laku) pengguna, guna menentukan ada dan tidaknya instrusi yang terjadi pada sistem. Namun demikian terdapat dua kondisi yang mana IDS salah dalam menentukan ada atau tidaknya intrusi, dua kondisi tersebut adalah *false positive* dan *false negative*. Hal tersebut yang melatarbelakangi banyaknya riset mengenai IDS. IDS mampu memberikan solusi pada berbagai jenis serangan namun IDS juga masih memiliki ketidakefektifan pada sistem kerjanya.

Sistem deteksi yang dibangun [14] terdiri dari empat komponen yakni WiFi *sniffer, frame filtering, frame characteristics and deauthentication detector, evil twin detection*. Penjelasan mengenai masing-masing komponen adalah sebagai berikut:

Gambar 1. Komponen sistem deteksi yang terdapat pada Skema Agarwal *et al.* [14]

a. WiFi *sniffer*

Fungsi utama dari bagian ini adalah untuk melakukan *sniffing frame* WiFi pada jaringan. *Sniffer* ini akan menangkap keseluruhan *frame* dari kanal WiFi yang sedang diawasi.

b. *Frame filtering*

Keseluruhan *frame* yang telah ditangkap akan disaring menggunakan modul ini. *Frame* yang ditujukan untuk *access point* lain akan dibuang dan tidak digunakan. Pada tahap ini dilakukan *scanning* proses *four-way handshake* antara *access point* yang ingin diawasi yang berupa *association response*.

Gambar 2. Format *association response frame*

Frame Control	Duration ID	Address 1 Client MAC	Address 2 BSSID (AP MAC)	Address 3 AP MAC	Sequence Control	Address 4	Network Data	FCS Checksum

Protocol Version	Type 0	Sub-Type 1	ToDS	FromDS	More Frag	Retry	Power Mgmt	More Data	WEP	Order

Format *association response frame* dijabarkan pada Gambar 2. Terdapat 3 parameter yang dipilih sebagai penentu ada atau tidaknya *evil twin attack*, yakni *retry bit, sequence number*, dan *association* ID. Ketiga hal tersebut dipilih karena memiliki karakteristik unik apabila terjadi perubahan nilai terhadap salah satu atau keseluruhan parameter, yang dapat dijadikan acuan dalam menentukan *evil twin attack*.

c. *Frame characteristics and deauthentication detector*

Apabila terdapat dua *association frame* yang diterima, maka diindikasi terdapat aktivitas yang mencurigakan, sedemikian sehingga pada tahap ini dilakukan pengecekan terhadap *frame characteristics* seperti *retry bit, sequence number,* dan *association* ID dari kedua respon guna mendeteksi adanya *evil twin attack*. Selain itu juga dilakukan pengecekan *deauthentication frame*.

d. *Evil twin detection*

Modul ini memberikan informasi kepada *administrator* mengenai terjadi atau tidaknya sebuah *evil twin attack*.

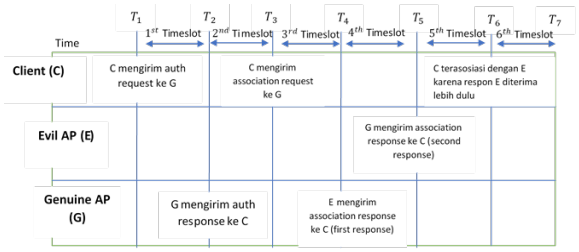
Berdasarkan penjabaran fungsi dari komponen-komponen tersebut, maka perlu diketahui bahwa *evil twin attack* diindikasikan terjadi apabila terdapat dua *association response* yang diterima klien pada proses *four-way handshake*.

2.2. *Evil Twin Attack*

Kelemahan protokol 802.11 mengundang berbagai macam serangan keamanan, seperti *denial of service, deauthentication attack, flooding attacks, rogue access point,* dan sebagainya [14]. Salah satu alternatif dari serangan-serangan tersebut adalah *evil twin attack* yang merupakan turunan dari *rogue access point*. Serangan dilakukan dengan mengecoh klien untuk mengasosiasikan perangkatnya pada suatu jaringan WiFi yang kemudian akan diarahkan pada suatu laman web yang telah disiapkan oleh penyerang guna mengambil informasi-informasi vital klien.

Penyerang dapat memanipulasi AP *evil twin* sedemikian sehingga menyerupai AP asli. Apabila sinyal AP *evil twin* lebih kuat daripada AP asli, maka perangkat klien hanya akan mendeteksi AP *evil twin*. Klien kemudian akan mengasosiasikan perangkatnya pada AP *evil twin*, dengan asumsi bahwa AP tersebut

memiliki koneksi internet. Cara kerja e*vil twin attack* diilustrasikan pada Gambar 3.



Gambar 3. Cara kerja *evil twin attack*

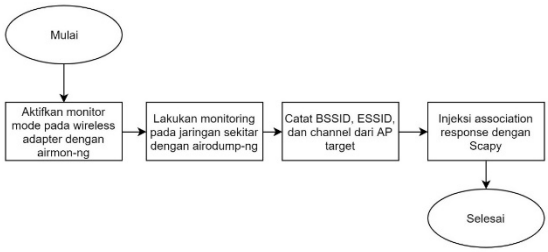
Metode yang digunakan klien dan juga *access point* untuk saling terasosiasi adalah metode *four-way handshake*. *Four-way handshake* harus dipenuhi agar dapat dilakukan kirim-terima data antara klien dan juga *access point*, proses ini diilustrasikan pada Gambar 4.

Gambar 4. Proses *four-way handshake*

Dalam proses *four-way handshake* terdapat kerawanan pada *management frame*. Celah kerawanan tersebut membuat banyak terjadinya serangan yang menargetkan *frames* yang termasuk jenis *management frame*, yakni *association response frame*. Salah satunya adalah *evil twin attack* yang dilakukan dengan menginjeksi parameter *association response frame*. Asumsi dari Gambar 4 adalah C yang merupakan klien ingin mengasosiasikan perangkatnya dengan AP yang seharusnya, G. Namun saat terjadi proses pengasosiasian, C tidak menyadari adanya AP penyerang, E, sedemikian sehingga *evil twin attack* dapat terjadi. Penyerang harus melakukan *spoofing* terhadap *authentication response frame* (proses *handshake* kedua), dan *spoofing* terhadap *association response frame* (proses *handshake* keempat). Cara kerja yang lebih detail adalah sebagai berikut:

a. *Timeslot* pertama (T_1) terjadi saat proses *handshake* pertama
   C mengirim *authentication request frame* ke G. E melakukan *sniffing* terhadap *authentication request* dari C.
b. *Timeslot* kedua (T_2) terjadi saat proses *handshake* kedua
   G mengirimkan balasan ke C berisi *authentication response*. C kemudian akan terautentikasi ke G. Proses autentikasi pada jaringan WiFi terbuka tidak melibatkan pertukaran kunci, hanya melibatkan adanya pertukaran *frame* antara klien dan AP. Pengiriman *authentication response* oleh E dapat membongkar adanya E dalam jaringan sehingga dapat terdeteksi oleh sistem deteksi, sedemikian sehingga E tidak akan mengautentikasi C dalam tahap ini. E hanya akan mengawasi dan mencatat tiap *parameter* yang dikirimkan oleh G kepada C ketika G mengirimkan *authentication response* ke C.
c. *Timeslot* ketiga (T_3) terjadi saat proses *handshake* ketiga
   C mengirimkan *association request frame* ke G.
d. *Timeslot* keempat (T_4) terjadi saat proses *handshake* keempat
   E mengirimkan *association response* ke C.
e. Timeslot kelima (T_5)
   G melakukan *sniffing* terhadap *association request* frame dari C yang dikirimkan pada T_3, sehingga G juga mengirimkan *association response* ke C.

Merujuk ke proses tersebut, dapat ditarik pernyataan bahwa hanya dengan menginjeksi satu *association response* saja sudah dapat dilakukan *evil twin attack*. *Evil twin attack* tidak perlu dilakukan menggunakan pengiriman *deauthentication frame* karena pengiriman *deauthentication frame* akan dideteksi sebagai *frame* yang dikirimkan ulang sedemikian sehingga tidak akan terdeteksi apabila dideteksi menggunakan sistem deteksi. Selanjutnya simulasi *evil twin attack* tersebut diilustrasikan pada Gambar 5.



Gambar 5. *Flowchart* simulasi *evil twin attack*

Simulasi *evil twin attack* diawali dengan mengaktifkan mode monitor pada *wireless adapter* yang dilakukan dengan *airmon-ng*. Hal ini bertujuan agar penyerang dapat melakukan *scanning* keseluruhan *traffic* yang ada pada jangkauan tertentu dengan menggunakan fitur *airodump-ng*. *Scanning* keseluruhan *traffic* dilakukan untuk mengetahui daftar AP yang berada di area tertentu. Daftar AP ini dapat dideteksi karena secara *default*, AP akan mengirimkan *beacon frames* ke alamat *broadcast* yang nantinya akan dideteksi sebagai AP oleh perangkat klien. Apabila AP target sudah terdaftar pada daftar *scan*, maka catat ESSID, BSSID, dan *channel*-nya yang kemudian akan dimanfaatkan untuk melakukan injeksi *association response* ke perangkat klien yang sudah terautentikasi dengan AP asli. Injeksi *association response* tersebutlah yang membuat perangkat klien menerima dua *association response* yang dijadikan dasar perbandingan pada sistem deteksi.

2.3. *Prototyping*

Systems Development Life Cycle (SDLC) merupakan sebuah metode untuk menentukan bagaimana suatu sistem dapat mendukung kebutuhan bisnis, mendesain sistem, merancang suatu sistem, dan menyampaikannya kepada pengguna. Pembuatan suatu sistem menggunakan SDLC harus melalui empat fase berupa *planning*, *analysis*, *design*, dan *implementation*. Setiap fasenya terdiri dari rangkaian tahapan yang berdasarkan pada teknik-teknik tertentu sedemikian sehingga hasil akhirnya dapat disampaikan kepada pengguna [19].

Pendekatan yang digunakan dalam metode ini adalah *system prototyping*. Pada pendekatan ini fase *analysis*, *design*, dan *implementation* dilakukan secara bersamaan sehingga sistem dapat diselesaikan dengan waktu yang relatif singkat. Hal tersebut ditujukan agar pengguna dapat memberikan umpan balik dan evaluasi. Pendekatan ini akan sangat berguna apabila pengguna memiliki kesulitan dalam menentukan kebutuhan bagi sistem.

Berikut merupakan tahapan-tahapan dalam *system prototyping*:

a. *Analysis*

Tahap ini berisi penjelasan mengenai sistem yang sudah ada sebelumnya, perbedaan atau perbaikan yang akan dilakukan dan pendataan mengenai kebutuhan-kebutuhan untuk sistem yang akan dibuat.

b. *Design*

Pada tahap ini akan dilakukan perancangan aplikasi untuk menentukan bagaimana sistem bekerja atau beroperasi dalam bentuk perangkat lunak.

c. *Implementation*

Pada tahap ini akan dilakukan konversi rancangan aplikasi menjadi *source code* yang kemudian akan diproses menjadi sebuah aplikasi atau perangkat lunak sebagai produk penelitian.

d. *System Prototype*

Versi sederhana dari sistem yang diusulkan akan dihasilkan pada tahap ini yang kemudian akan dilakukan perbaikan kembali jika sistem tersebut masih belum sesuai atau belum memenuhi kebutuhan.

e. *System*

Pada tahap ini sistem sudah selesai dibuat.

2.4. *Penelitian Terkait*

Adanya *evil twin attack* yang perkembangannya semakin variatif sedikit banyak melatarbelakangi para peneliti untuk melakukan riset lebih lanjut guna mendeteksi atau mengatasi serangan ini pada jaringan nirkabel. Adapun solusi-solusi yang pernah dipublikasikan berdasarkan metode yang digunakan adalah sebagai berikut:

a. *Monitoring wireless traffic*

Solusi-solusi yang menggunakan metode ini menitikberatkan penggunaan *sniffer* untuk digunakan sebagai *tools* guna mendapatkan informasi-informasi penting seperti *MAC address* dari *access point*, SSID, nilai RSSI, *channel*, dan sebagainya. Informasi-informasi penting inilah yang nantinya digunakan sebagai parameter pendeteksi *evil twin attack*. Kemudian dipublikasikan usulan mengenai penggunaan *white list* yang berisi *MAC address* dari *AP* asli [16]. Apabila pada suatu waktu *sniffer* mendapati bahwa terdapat AP yang MAC *address*-nya tidak terdapat di *white list* maka sistem akan melakukan *tracing* pada lokasi AP. Selanjutnya, apabila lokasi AP tidak sesuai dengan kondisi ideal maka akan terdeteksi adanya *evil twin attack*. Sriram dan Chirumamilla mengusulkan penggunaan *agent-based* IDS dalam melakukan deteksi *evil twin attack* [17], [18]. IDS berfungsi untuk melakukan *monitoring* jaringan dan mendeteksi adanya AP baru. Apabila terdapat AP baru yang tidak terdapat pada daftar yang telah dideklarasikan maka AP tersebut akan ditandai sebagai *evil twin* AP.

b. *Feature extraction and timing-based solution*

Metode yang digunakan adalah dengan mengumpulkan seluruh *frame* menggunakan sensor nirkabel yang kemudian dilakukan analisis lebih dalam terhadap *frame-frame* tersebut. Informasi yang didapatkan dari keseluruhan *frame* kemudian digunakan untuk menentukan ada atau tidaknya *evil twin attack*. Selain itu, adanya asumsi bahwa kebanyakan *evil twin* AP dibuat dengan menggunakan koneksi *bridge* untuk menyediakan koneksi internet, maka metode ini juga berfokus pada jumlah waktu yang dibutuhkan klien untuk terkoneksi terhadap *access point.* Hal ini didasari karena apabila koneksi yang digunakan *evil twin* AP adalah koneksi *bridge* dari AP asli, maka akan terdapat *additional hop* yang pada akhirnya akan menyebabkan *delay*. Han *et al.* mengusulkan penggunaan metode berbasis waktu untuk mendeteksi adanya *evil twin attack* yang mana memanfaatkan fungsi penghitungan *Round Trip Time* (RTT) untuk DNS *query* [11]. Kemudian Song *et al.* juga mengusulkan penggunaan metode berbasis waktu berupa *inter arrival time* (IAT) untuk mendapatkan *delay* yang disebabkan oleh adanya *evil twin* AP [13].

c. *Proprietary hardware*

Metode yang ketiga adalah penggunaan perangkat keras yang dimanfaatkan untuk mengirimkan pesan kepada seluruh perangkat klien yang telah terasosiasi dengan *access point*, yang mana pesan tersebut berisi instruksi agar seluruh klien tidak merespon *probe request* yang akan dikirimkan oleh AP [8]. Dengan demikian, apabila terdapat AP yang membalas *probe request* dapat ditarik kesimpulan bahwa AP tersebut termasuk ke dalam *evil twin* AP.

Beberapa solusi memang telah diusung namun masih saja terdapat kekurangan-kekurangan pada solusi-solusi yang telah dipublikasikan. Penggunaan daftar AP asli seperti yang dilakukan pada metode *monitoring wireless traffic* tidak dapat diterapkan secara ideal karena pada dasarnya *evil twin* AP dibuat

dengan melakukan *spoofing* SSID dan BSSID dari AP asli sedemikian sehingga MAC asing tidak akan terdeteksi karena sama dengan yang ada pada daftar.

Kemudian, penggunaan metode *feature extraction and timing-based solution* juga masih memiliki kekurangan, yang mana besar kemungkinan penyerang untuk menyediakan koneksi internetnya sendiri—tidak menggunakan koneksi *bridging* dengan AP asli. Hal tersebut membuat tidak adanya *delay* yang terdeteksi. Di samping itu, *delay* yang muncul tidak hanya berasal dari anggapan adanya *evil twin* AP. *Delay* bisa juga disebabkan oleh *router* yang sedang mengalami *bottleneck* sehingga banyak *frame* yang belum terkirim. Selain itu, metode ini juga masih rentan untuk diaplikasikan karena untuk mendeteksi *evil twin attack* tidak semata-mata dapat diukur dari selisih waktu yang dibutuhkan untuk melakukan transmisi. Penyerang yang sadar akan adanya *delay* transmisi akan memilih untuk menginjeksi respon yang telah di-*spoof* kepada klien yang mana hal tersebut tidak akan terdeteksi oleh metode *timing-based*.

Selanjutnya, pemanfaatan *proprietary hardware* juga masih memiliki kekurangan. Hal ini berkaitan dengan apa yang telah disebutkan di atas, perangkat klien yang tidak mengirimkan balasan untuk *probe request* yang dikirim oleh suatu perangkat merupakan suatu tindakan yang menyalahi standard IEEE 802.11. Selain itu, tanpa mengetahui skenario, penyerang juga dapat memilih untuk tidak mengirimkan balasan kepada perangkat keras tersebut, yang nantinya penyerang tidak akan terdeteksi sebagai *evil twin* AP. Selain penggunaan *proprietary hardware*, penggunaan IDS juga banyak diusulkan. Meskipun demikian, *signature-based* dan *anomaly-based* IDS belum dapat mendeteksi *evil twin attack* dengan sempurna. Hal tersebut didasari karena *signature-based* IDS hanya akan mendeteksi apa-apa saja yang sudah di-*input* ke dalam *database*, sedangkan *evil twin attack* bersifat dinamis sedemikian sehingga sulit untuk selalu melakukan *update database*. Kemudian ketidakefektifan pada *anomaly-based* IDS adalah akan muncul banyaknya *false positive* karena *evil twin attack* merupakan suatu hal yang dapat diubah-ubah parameternya kapan saja oleh penyerang.

3. METODOLOGI PENELITIAN

Metodologi penelitian terdiri dari tahapan-tahapan yang selanjutnya akan digunakan oleh peneliti sebagai sarana untuk mempelajari masalah penelitian [20]. Metodologi penelitian yang akan digunakan pada penelitian ini adalah metode *Design Research Methodology* (DRM). Metode DRM dipilih sebagai metodologi karena dapat digunakan untuk melakukan pendekatan yang lebih teliti sehingga tujuan untuk mendapat desain yang lebih efektif dan efisien dapat tercapai [21]. Selain metodologi penelitian, adanya pengembangan aplikasi memungkinkan untuk menggunakan metode pengembangan aplikasi atau perangkat lunak. Metode yang digunakan berupa metode *System Development Life Cycle* (SDLC) dengan pendekatan *prototyping.* SDLC akan masuk dalam tahapan *Prescriptive Study* guna sebagai penyelesaian masalah dalam penelitian.

Terdapat 4 tahapan pada metode DRM yang digunakan yakni berupa *Research Clarification, Descriptive Study* I, *Prescriptive Study*, dan *Descriptive Study* II.

4. IMPLEMENTASI SISTEM DETEKSI SERANGAN *EVIL TWIN*

4.1. Proses Pembangunan Prototipe

Pembangunan prototipe dilakukan dengan metode SDLC dan dengan pendekatan *prototyping.* Seperti yang telah disebutkan sebelumnya bahwa metode ini terdiri dari empat tahapan yang berupa *planning*, *analysis*, *design*, dan *implementation.* Tahap *planning* telah dilakukan saat menentukan tujuan yang akan dicapai, yakni tersedianya prototipe sistem deteksi untuk mendeteksi *evil twin attack* pada jaringan nirkabel. Selanjutnya adalah tahap analisis yang berupa analisis kebutuhan yang terdiri dari kebutuhan fungsional dan non-fungsional sistem. Analisis kebutuhan dibuat berdasarkan pada IEEE 830-1998 tentang dokumen *Software Requirement Specification* (SRS) yang selanjutnya kebutuhan fungsional dan non-fungsionalnya merujuk pada rujukan utama dibuatnya penelitian ini. Adapun kebutuhan fungsional dari penelitian ini adalah:

a. Sistem deteksi dapat mendeteksi *adanya evil twin attack* dengan membandingkan *association response*.
b. Sistem deteksi memiliki kemampuan untuk *sniffing* dan *filtering* paket data pada *monitor mode*.
c. Sistem deteksi melakukan *sniffing* pada alamat MAC yang telah didefinisikan sebelumnya.
d. Sistem deteksi yang dibangun adalah sistem deteksi *real-time*.
e. Sistem deteksi memberikan *warning* apabila terdeteksi adanya *evil twin attack.*

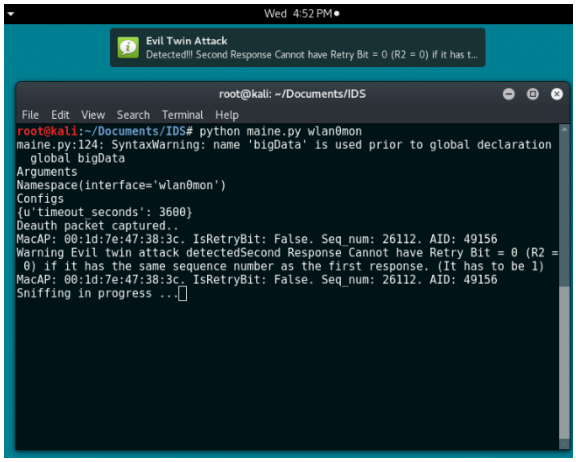
Tahapan selanjutnya adalah perancangan sistem. Perancangan sistem dilakukan dengan menggunakan *flowchart*. *Flowchart* memodelkan mengenai cara kerja sistem deteksi dan deteksi *evil twin attack.* Algoritme deteksi *evil twin attack* dijabarkan dengan pseudocode pada Gambar 6.

Lalu selanjutnya adalah mengimplementasikan desain sistem yang telah dirancang menjadi suatu sistem yang sebenarnya. Pembangunan sistem deteksi dilakukan dengan bahasa pemrograman Python dan dengan memanfaatkan beberapa *tools* dan *library* tambahan. Di antaranya adalah Scapy untuk melakukan *sniffing*, *regex* untuk melakukan *filtering*, *pandas* untuk membuat *log*, dan *notify2* untuk memunculkan *warning sign*.

```
r1 = retry bit of first response
r2 = retry bit of second response
seq1 = sequence number of first response
seq2 = sequence number of second response
aid1 = AID of first response
aid2 = AID of second response
deauth = deauth frame

if deauth
    elif r1 == 0
        store r1, seq1, aid1 live
        if r2 == 0
            store r2, seq2, aid2 live

            if seq1 == seq2
                print "Evil Twin Detected"
            else
                if deauth
                    if seq1 == seq2
                    print "Evil Twin Detected"
        if r2 == 1
            store r2, seq2, aid2 live

            if seq1 == seq2
                if aid1 != aid2
                    print "Evil Twin Detected"
            else
                print "Evil Twin Detected"
        else
            normal condition, print "Normal
Condition"
    else
        r1 == 1
        store r1, seq1, aid1 live

        if r2 == 0
            store r2, seq2, aid2 live

            if seq1 == seq2
                print "Evil Twin Detected"

            else
                if deauth
                    if seq1 == seq2
                    print "Evil Twin Detected"

        if r2 == 1
            store r2, seq2, aid2 live

            if seq1 == seq2
                if aid1 != aid2
                    print "Evil Twin Detected"
            else
                print "Evil Twin Detected"
        else
            normal condition, print "Normal
Condition"
back to sniff
```

Gambar 6. *Pseudocode* cara kerja sistem deteksi

Hasil akhir prototipe sistem deteksi saat mendeteksi serangan *evil twin* disematkan pada Gambar 7.

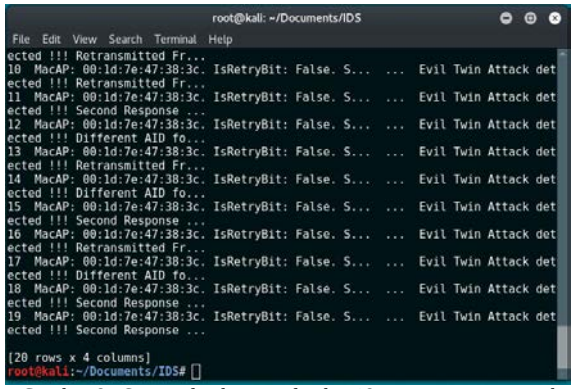


Gambar 7. Sistem deteksi saat mendeteksi adanya evil twin attack

4.2. Hasil Pengujian Sistem Deteksi

Pengujian terhadap prototipe sistem deteksi dilakukan sebanyak dua kali yang terdiri dari pengujian sistem dan juga pengujian terhadap serangan. Pengujian sistem terdiri dari *unit testing*, *integration testing*, dan *system testing*. Pengujian menunjukkan bahwa sistem deteksi yang dibangun telah dapat berfungsi sesuai dengan perancangan aplikasi yang telah dilakukan. Hal ini ditunjukkan dengan hasil dari tiap-tiap uji. Pada *unit testing* seluruh komponen telah memenuhi keseluruhan spesifikasi dan berjalan sesuai fungsi, selanjutnya seluruh hubungan kolaborasi antara komponen-komponen pada aplikasi juga sudah diperiksa dan sesuai dengan spesifikasi yang sebelumnya telah diuji menggunakan *integration testing*, kemudian pada *system testing*, uji menghasilkan pernyataan bahwa aplikasi telah memenuhi seluruh kebutuhan fungsional dan non-fungsional.

Adapun pengujian terhadap serangan dilakukan dengan percobaan *evil twin attack* terhadap sistem deteksi selama satu jam. Dalam rentang waktu satu jam tersebut sistem deteksi dapat mendeteksi keseluruhan serangan yang dilakukan yakni sebanyak 18 kali serangan secara *real-time* yang diilustrasikan pada Gambar 8.



Gambar 8. Sistem deteksi mendeteksi 18 serangan secara *real-time*

Injeksi *association response* yang dikirimkan oleh penyerang adalah sejumlah 18 *frame*. Hal ini membuktikan bahwa sistem deteksi mampu menangkap keseluruhan *frame* yang diinjeksi dan membandingkannya secara tepat sehingga mampu mendeteksi *evil twin attack* pada keseluruhan injeksi *frame* yang dikirimkan. Ketepatan deteksi ditunjukkan dengan hasil keseluruhan *association response* yang disimpan pada *log* dan juga *warning* yang ditampilkan pada *notification panel*. Sistem deteksi bekerja secara *real-time*, dibuktikan dengan waktu injeksi dan waktu *association response* diterima dan dideteksi sebagai *evil twin attack* yang sama.

5. SIMPULAN DAN SARAN

Berdasarkan hasil penelitian, dapat disimpulkan bahwa sistem deteksi dapat memenuhi kebutuhan

fungsional dan non-fungsional serta dapat memenuhi uji keamanan yang dibuktikan bahwa dapat terdeteksinya keseluruhan serangan *evil twin attack* yang dilakukan sebanyak 18 kali serangan dalam kurun waktu satu jam. Hasil penelitian juga menunjukkan bahwa fungsi sistem deteksi ini memang hanya dibatasi hingga pendeteksian serangan, dan tidak mencakup fungsi penghentian serangan

Saran untuk pengembangan selanjutnya adalah di antaranya:

a. Penelitian ini dapat dilanjutkan dengan penggunaan metode lain untuk mendeteksi *evil twin attack*.
b. Penelitian ini dibatasi pada pendeteksian *evil twin attack*, sehingga untuk penelitian selanjutnya bisa ditambahkan fungsi untuk menghentikan serangan.

Referensi

[1] R. Di Pietro and G. Oligeri, "Silence is Golden: Exploiting Jamming and Radio Silence to Communicate," *ACM Trans. Inf. Syst. Secur.*, vol. 17, no. 3, pp. 1–24, 2015.

[2] Y. Gilad and A. Herzberg, "Off-Path TCP Injection Attacks," *ACM Trans. Inf. Syst. Secur.*, vol. 16, no. 4, pp. 1–32, 2014.

[3] J. Dong, R. Curtmola, and C. Nita-Rotaru, "Practical defenses against pollution attacks in wireless network coding," *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 1, pp. 1–31, 2011.

[4] J. Bellardo and S. Savage, "802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions.," *USENIX Secur.*, pp. 15–28, 2003.

[5] C. Anagnostopoulos, "Intelligent Contextual Information Collection in Internet of Things," *Int. J. Wirel. Inf. Networks*, vol. 23, no. 1, pp. 28–39, 2016.

[6] W. Wei *et al.*, "Passive online detection of 802.11 traffic using sequential hypothesis testing with TCP ACK-pairs," *IEEE Trans. Mob. Comput.*, vol. 8, no. 3, pp. 398–412, 2009.

[7] L. Ma, A. Y. Teymorian, X. Cheng, and M. Song, "RAP: protecting commodity wi-fi networks from rogue access points," *Fourth Int. Conf. Heterog. Netw. Qual. Reliab. Secur. Robustness Work. - QSHINE '07*, p. 1, 2007.

[8] P. K. Dubey and J. N. Verma, "(12) Patent Application Publication (10) Pub. No.: US 2012/0124665 A1," vol. 1, no. 19, 2012.

[9] J. Copeland, S. Kangude, R. Beyah, George Yu, and B. Strickland, "Rogue access point detection using temporal traffic characteristics," pp. 2271–2275, 2005.

[10] P. Chumchu, T. Saelim, and C. Sriklauy, "A new MAC address spoofing detection algorithm using PLCP header," *Int. Conf. Inf. Netw. 2011, ICOIN 2011*, pp. 48–53, 2011.

[11] H. Han, B. Sheng, C. C. Tan, Q. Li, and S. Lu, "A timing-based scheme for rogue AP detection," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 11, pp. 1912–1925, 2011.

[12] Q. Liao *et al.*, "RIPPS: Rogue identifying packet payload slicer detecting unauthorized wireless hosts through network traffic conditioning.," *ACM Trans. Inf. Syst. Secur.*, vol. 11, no. 2, pp. 1–23, 2008.

[13] Y. Song, C. Yang, and G. Gu, "Who is peeping at your passwords at starbucks? - To catch an evil twin access point," *Proc. Int. Conf. Dependable Syst. Networks*, pp. 323–332, 2010.

[14] M. Agarwal, S. Biswas, and S. Nandi, "An Efficient Scheme to Detect Evil Twin Rogue Access Point Attack in 802.11 Wi-Fi Networks," *Int. J. Wirel. Inf. Networks*, vol. 25, no. 2, pp. 130–145, 2018.

[15] R. Shirey, "RFC 2828-Internet security glossary." 2003.

[16] K. Kao, T. Yeo, W. Yong, and H. Chen, "A Location-aware Rogue AP Detection System Based on Wireless Packet Sniffing of Sensor APs," pp. 32–36, 2011.

[17] V. S. S. Sriram and G. Sahoo, "Detecting and Eliminating Rogue Access Points in IEEE-802 . 11 WLAN - A Multi-Agent Sourcing Methodology," pp. 256–260, 2010.

[18] M. K. Chirumamilla, "Agent Based Intrusion Detection and Response System for Wireless LANs," pp. 492–496, 2003.

[19] A. Dennis, B. H. Wixom, and R. M. Roth, *System Analysis and Design 5th Edition*, 5th ed. John Wiley & Sons, Inc., 2012.

[20] K. Nallaperumal, "Engineering research methodology A computer science and engineering and information and communication technologies perspective," no. December 2013, 2015.

[21] L. T. M. Blessing and A. Chakrabarti, *DRM, a design research methodology*. 2009.

Serangan *Impossible Differential* Pada *Reduced Round* SKINNY Berbasis Teknik *Miss-In-The-Middle*

Daniel Pascah Pasaribu[1)] dan Santi Indarjani[2)]
(1) Sekolah Tinggi Sandi Negara / daniel.pascah@student.poltekssn.ac.id
(2) Sekolah Tinggi Sandi Negara / santi.indarjani@poltekssn.ac.id

Abstrak

SKINNY adalah algoritme yang didesain oleh Beierle et al.dengan struktur Substitution-Permutation Network *(SPN). Beierle et al. melakukan beberapa serangan untuk menganalisis keamanan dari SKINNY, salah satunya yaitu serangan* impossible differential *pada 16 round SKINNY. Pada penulisan ini dilakukan rekonstruksi serangan* impossible differential *pada SKINNY menggunakan teknik* miss-in-the-middle *dengan 4 skenario* forward *dan* backward differential path *untuk semua kombinasi satu dan dua* active nibble*. Pencarian* impossible differential path *dilakukan dengan mencari* forward *dan* backward differential path *hingga diperoleh* round *terpanjang.* Path *dengan* round *terpanjang yang diperoleh dari semua kemungkinan kombinasi pada* forward *dan* backward *untuk membentuk* impossible differential path *terbaik. Analisis serangan dilakukan dengan memperhatikan* path *dan pengaruh antara input dengan* output difference *serta penghitungan kompleksitas data dengan memperluas* round *ke atas dan ke bawah pada* impossible differential path*. Penelitian ini juga melakukan penebakan kunci (subtweakey) menggunakan 4 skenario berdasarkan informasi yang diperoleh dari perluasan* round*. Hasil penelitian menunjukkan bahwa terdapat* impossible differential path reduced round *terbaik yaitu pada skenario 1 yang dapat membentuk* impossible differential path *hingga 11* round *seperti yang ditemukan Beierle et al. (2016). Dari hasil percobaan penebakan kunci, skenario ini membutuhkan kompleksitas data sebanyak* $2^{41,5}$ plaintext *untuk menebak 32 bit* subtweakey *dengan perluasan* round *hingga 16.*

Kata Kunci: *serangan impossible differential* (1)*, karakteristik impossible differential path* (2)*, kompleksitas data* (3)*, SKINNY* (4)*, miss-in-the-middle (5)*

1. PENDAHULUAN

SKINNY merupakan salah satu algoritme *lightweight cryptography* dengan struktur *Substitution-Permutation Network* (SPN). SKINNY didesain oleh Beierle *et al.* (2016) dan memiliki enam varian yaitu SKINNY-*n*-*n*, SKINNY-*n*-*2n*, dan SKINNY-*n*-*3n* dengan *n* merupakan ukuran *input* algoritme ($n = 64$ dan 128 bit). Operasi-operasi yang terdapat pada SKINNY adalah *SubCells*, *AddConstants*, *AddRoundTweakey*, *ShiftRows*, dan *MixColumns* [1]. SKINNY memiliki "*tweakey*" sebagai *input* dengan tujuan untuk meningkatkan performa algoritme dengan biaya yang rendah [2]. Oleh karena itu, SKINNY juga merupakan algoritme *tweakable block cipher*.

Terdapat beberapa metode untuk melakukan serangan, salah satunya adalah serangan *impossible differential*, yaitu sebuah serangan untuk menghubungkan dua *differential path* dengan sebuah kontradiksi, sedemikian sehingga *differential* tersebut tidak pernah terjadi [3]. Sebelum dilakukan *recovery* kunci pada penerapan serangan *impossible differential*, diperlukan sebuah *impossible differential path* yang dapat mencakup jumlah *round* sebanyak mungkin. *Impossible differential path* dapat dibangun dengan teknik *miss-in-the-middle* [4]. Beberapa upaya serangan yang dilakukan oleh Beierle *et al.* (2016) untuk menganalisis keamanan dari SKINNY, antara lain serangan *differential* dan *linear*, serangan *meet-in-the-middle*, serangan *impossible differential*, serangan integral, *slide attack*, serangan *subspace*, dan *algebraic attack*. Beierle *et al*. (2016) melakukan serangan *impossible differential* dengan skenario satu *active nibble* pada *forward* dan *backward differential path*. Jumlah *round* maksimum dari *impossible differential path* algoritme SKINNY yang diperoleh adalah sebanyak 11 *round*. Untuk mengetahui ketahanan SKINNY terhadap serangan *impossible differential* maka pada penelitian ini dilakukan penerapan *impossible differential* pada *reduced round* SKINNY. Pencarian karakterisik *impossible differential* dilakukan menggunakan teknik *miss-in-the-middle* dengan empat skenario *forward* dan *backward differential path* berdasarkan kombinasi satu dan dua *active nibble*. Pencarian karakteristik *impossible differential path* dilakukan dengan mencari *forward* dan *backward differential path* hingga diperoleh *round* terpanjang. Parameter *round* terpanjang yaitu *round* dengan kondisi masih terdapat *active nibble* atau *passive nibble*. Selanjutnya, penelitian ini juga menghitung kompleksitas data yang dibutuhkan untuk melakukan serangan *impossible differential* pada SKINNY.

2. LANDASAN TEORI

2.1. *Lightweight Block Cipher*

Lightweight cryptography adalah algoritme atau protokol kriptografi yang dirancang agar mampu diimplementasikan ke dalam suatu perangkat yang memerlukan sumber daya yang rendah, seperti memori dan daya komputasi yang terbatas [5]. *Lightweight cryptography* dapat diklasifikasikan ke

dalam *lightweight block cipher*, *lightweight hash function*, dan *lightweight public key cryptography*. *Block cipher* adalah sebuah fungsi yang memetakan *n* bit blok *plaintext* menjadi *n* bit blok *ciphertext* [6]. *Lightweight block cipher* dapat diartikan sebagai suatu fungsi kriptografi yang memetakan *n* bit *input plaintext* ke-*n* bit *output ciphertext* yang dapat diimplementasikan pada perangkat dengan sumber daya yang terbatas [7]. SKINNY (Beierle *et al*, 2016) merupakan salah satu contoh dari *lightweight block cipher*.

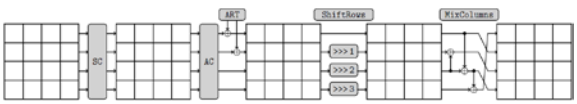
2.2. Algoritme SKINNY

Penjelasan mengenai algoritme SKINNY-64-64 seluruhnya merujuk pada [1]. SKINNY adalah algoritme *lightweight tweakable block cipher* yang diajukan oleh Beierle *et al.* pada CRYPTO 2016. Algoritme ini memiliki struktur *Substitution-Permutation Network* (SPN) dengan dua jenis varian ukuran blok n yaitu 64 dan 128 bit. SKINNY menggunakan kerangka *tweakey* seperti konstruksi yang diajukan oleh Jean *et al*. pada tahun 2014 [8]. Algoritme *tweakable block cipher* dengan kerangka *tweakey* yaitu algoritme yang mengganti kunci masukan atau pasangan kunci/*tweak* dengan menggunakan *tweakey* (TK). *Tweakey* pada algoritme SKINNY terdiri dari tiga varian ukuran, yaitu $t = n$, $t = 2n$, dan $t = 3n$. Varian algoritme SKINNY dapat dituliskan dengan SKINNY-n-t. Pada penelitian ini, varian algoritme SKINNY yang digunakan memiliki ukuran blok dan *tweakey* 64 bit atau dituliskan dengan SKINNY-64-64.

Pada SKINNY-64-64, *plaintext* direpresentasikan sebagai $X = x_0 \parallel x_1 \parallel \cdots \parallel x_{14} \parallel x_{15}$, kemudian dimasukkan ke dalam *internal state X* seperti pada matriks berukuran 4×4 berikut:

$$X = \begin{bmatrix} x_0 & x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 & x_7 \\ x_8 & x_9 & x_{10} & x_{11} \\ x_{12} & x_{13} & x_{14} & x_{15} \end{bmatrix}.$$

Proses enkripsi SKINNY-64-64 menerapkan *round function* yang diiterasi sebanyak 32 *round*. Proses dekripsi dilakukan dengan menggunakan invers dari *round function*. *Round function* pada SKINNY-64-64 terdiri dari lima operasi dengan urutan *SubCells*, *AddConstants*, *AddRoundTweakey*, *ShiftRows*, dan *MixColumns*, seperti yang ditunjukkan pada Gambar 1.



Gambar 1. *Round function* pada SKINNY-64-64 [3]

Fungsi *SubCells* (*SC*) memetakan *X* secara bijektif oleh *s-box* seperti yang ditunjukkan pada Tabel 1. Pada fungsi *AddConstants* (*AC*), *X* di-XOR-kan dengan *round function* algoritme SKINNY-64-64 (Lihat Tabel 2). *Subtweakey* di-XOR-kan dengan *X* pada proses *AddRoundTweakey* (*ART*). Proses enkripsi dilanjutkan dengan fungsi *ShiftRows* yang merotasi *X* dengan ketentuan seperti ditunjukkan pada Gambar 2. *Round function* terakhir yaitu *MixColumns* yang mengalikan setiap kolom pada *X* dengan matriks biner *M* sebagai berikut:

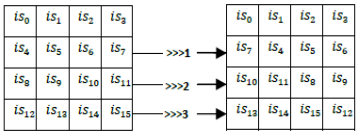
$$M = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}.$$

Tabel 1. *S-box* dari SKINNY-64-64 [1]

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
s	c	6	9	0	1	a	2	b	3	8	5	d	4	e	7	f

Tabel 2. Konstanta putaran SKINNY-64-64 [1]

Round	Konstanta (heksadesimal)
1-16	01,03,07,0f,3e,3d,3b,37,2f,1e,3c,39,33,27,0e
17-32	1d,3a,35,2b,16,2b,18,30,21,02,05,0b,17,2e,1c



Gambar 2. Fungsi *ShiftRows* pada SKINNY-64-64

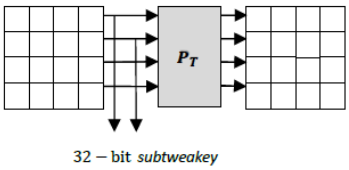
Subtweakey pada SKINNY-64-64 dihasilkan oleh algoritme *tweakey schedule* dengan *input* berupa *tweakey* berukuran 64 bit. Masukan *tweakey* dinotasikan sebagai $TK = tk_0 \parallel tk_1 \parallel \cdots \parallel tk_{15}$ yang direpresentasikan dalam bentuk matriks 4×4

$$TK = \begin{bmatrix} tk_0 & tk_1 & tk_2 & tk_3 \\ tk_4 & tk_5 & tk_6 & tk_7 \\ tk_8 & tk_9 & tk_{10} & tk_{11} \\ tk_{12} & tk_{13} & tk_{14} & tk_{15} \end{bmatrix}.$$

Algoritme ini menggunakan *array* $TK1$ untuk membentuk *subtweakey* karena memiliki nilai *z* sama dengan satu $\left(z = \frac{t}{n} = \frac{64}{64} = 1\right)$. Notasi $TK1^r = tk1_0^r, \ldots, tk1_{15}^r$ untuk $1 \le r \le 32$ menyatakan *array* $TK1$ pada *round* r. Nilai awal dari *array* $TK1^1 = TK$. Proses *tweakey schedule* untuk $1 \le r \le 32$ dilakukan sebagai berikut:

a. *subtweakey* pada *round* ke-r adalah $tk1_0^r, \ldots, tk1_7^r$;
b. $TK1^r$ diperbarui menggunakan fungsi permutasi $P_T = [9,15,8,13,10,14,12,11,0,1,2,3,4,5,6,7]$ untuk *round* selanjutnya atau dengan kata lain $TK1^{r+1} = P_T(TK1^r)$.

Gambar 3 menunjukkan proses *tweakey schedule* pada SKINNY-64-64.



Gambar 3. T*weakey* schedule SKINNY-64-64 [3]

2.3. Serangan *Impossible Differential*

Serangan *impossible differential* adalah serangan yang menggunakan *input* dan *output difference* dengan menghubungkan dua *differential path* untuk menemukan kunci yang benar dari nilai probabilitas yang besar dan membuang kunci yang kontradiksi, maka *differential* tersebut tidak pernah terjadi [3]. Serangan *impossible differential* merupakan pengembangan dari serangan *differential* dengan mencari probabilitas nol, yaitu *differential* yang tidak mungkin terjadi atau *impossible* [9]. *Miss-in-the-middle* merupakan salah satu teknik yang digunakan dalam melakukan pencarian karakteristik *impossible differential* [4]. Serangan *impossible differential* terdiri dari dua langkah utama. Pertama, berkaitan dengan pencarian *round impossible differential* terpanjang, yaitu *input difference* ΔX dan *output difference* ΔY sedemikian sehingga probabilitas ΔX, setelah sejumlah *round*, menghasilkan *output difference* yang bernilai nol. Suatu karakteristik *impossible differential* dapat dikatakan baik apabila dapat mencakup jumlah *round* sebanyak mungkin, sehingga dapat dilanjutkan ke langkah kedua yaitu *recovery* kunci. *Truncated differential* digunakan untuk membentuk *impossible differential path* yang dapat dikembangkan menjadi karakteristik *impossible differential* [10].

2.4. Teknik *Miss-In-The-Middle*

Teknik *miss-in-the-middle* pada serangan *impossible differential* menghubungkan dua *differential path* dengan probabilitas satu sehingga terdapat kontradiksi yang ditemukan. Berikut merupakan langkah dasar dari teknik *miss-in-the-middle*:

a. Menentukan *input difference* ΔX dari algoritme.
b. Mendapatkan seluruh kemungkinan ΔZ_r , yaitu *differences* pada *round* ke-r dari input *difference* ΔX.
c. Menemukan himpunan posisi bit dari *difference* ΔZ_r yang nilainya selalu *zero* (atau *nonzero*). Jika tidak ditemukan himpunan seperti itu, kembali ke langkah a). Jika tidak ada posisi bit dari *difference* ΔZ_r yang nilainya selalu tetap untuk semua *input difference*, maka tidak terdapat *impossible differential* untuk *cipher* tersebut.
d. Memilih *output difference* ΔY dari algoritme.
e. Mencari seluruh kemungkinan $\Delta Z_r'$ yaitu *differences* pada *round* ke-r (*round* yang sama seperti pada langkah b.) dari *output difference*.
f. Mengecek kembali apakah nilai pada posisi yang sama seperti pada langkah c. dari *difference* ΔZ_r selalu *nonzero* (atau *zero*) atau dengan kata lain memiliki nilai *difference* yang selalu berbeda pada ΔZ_r dan $\Delta Z_r'$ di posisi bit yang bersesuaian. Jika kondisi tersebut terpenuhi, maka telah ditemukan sebuah *impossible differential* dengan *input difference* ΔX dan *output difference* ΔY. Mengulangi langkah d. untuk kemungkinan *output differences* yang lain jika kondisi tersebut tidak terpenuhi.

2.5. Sifat XOR terhadap *Nibble Difference*

Pengaruh operasi XOR terhadap *nibble difference* dibagi dalam enam sifat, yaitu Sifat 1, Sifat 2, dan Sifat 3 [7] dilanjutkan dengan Sifat 4, Sifat 5, dan Sifat 6 [11]. Berikut merupakan pengaruh operasi XOR terhadap *nibble difference*.

Sifat 1. *Passive nibble* di-XOR-kan dengan *passive nibble* menghasilkan *passive nibble*.
Bukti.
Misal Δx dan Δy merupakan *passive nibble*, selanjutnya dibuktikan bahwa Δz merupakan *passive nibble*.

$$\begin{array}{ccccc} \Delta x & \oplus & \Delta y & = & \Delta z \\ 0 & \oplus & 0 & = & \Delta z \\ & & 0 & = & \Delta z \end{array}$$

Terbukti bahwa Δz merupakan *passive nibble*.

Sifat 2. *Passive nibble* di-XOR-kan dengan *active nibble* (dan sebaliknya) menghasilkan *active nibble*.
Bukti.
Misal Δx merupakan *passive nibble* dan Δy merupakan *active nibble*, selanjutnya dibuktikan bahwa Δz merupakan *active nibble*.

$$\begin{array}{ccccc} \Delta x & \oplus & \Delta y & = & \Delta z \\ 0 & \oplus & \alpha & = & \Delta z \\ & & \alpha & = & \Delta z \end{array}$$

Terbukti bahwa Δz merupakan *active nibble*.

Sifat 3. *Active nibble* di-XOR-kan dengan *active nibble* menghasilkan *otherwise nibble*.
Bukti.
Misal Δx dan Δy merupakan *active nibble*, selanjutnya dibuktikan bahwa Δz merupakan *otherwise nibble*.

$$\begin{array}{ccccc} \Delta x & \oplus & \Delta y & = & \Delta z \\ \gamma & \oplus & \gamma' & = & \Delta z \end{array}$$

Pada persamaan di atas, terdapat dua kondisi dalam menentukan *output* dari $\gamma \oplus \gamma'$, yaitu

1. Apabila $\gamma = \gamma'$, maka $\Delta z = 0$

$$\begin{array}{ccccc} \gamma & \oplus & \gamma' & = & \Delta z \\ \gamma' & \oplus & \gamma' & = & \Delta z \\ & & 0 & = & \Delta z \end{array}$$

2. Apabila $\gamma \neq \gamma'$, maka $\Delta z \neq 0$

$$\begin{array}{ccccc} \gamma & \oplus & \gamma' & = & \Delta z \\ \gamma & \oplus & \gamma' & = & \Delta z \\ & & \alpha & = & \Delta z \end{array}$$

Terbukti bahwa Δz merupakan *otherwise nibble*.

Sifat 4. *Active nibble* di-XOR-kan dengan *otherwise nibble* (dan sebaliknya) menghasilkan *otherwise nibble*.
Bukti.
Misal Δx merupakan *active nibble*, Δy merupakan *otherwise nibble*, dan Δz adalah

hasil XOR antara Δx dan Δy. Terdapat dua kemungkinan kondisi untuk Δy, yakni bersifat *active* atau *passive*. Jika Δy bersifat *active*, maka Δz bersifat *otherwise* berdasarkan sifat 3. Apabila Δy bersifat *passive*, maka berdasarkan sifat 2, Δz bersifat *active*. Oleh karena itu, Δz bersifat *otherwise*. Demikian pula jika Δx merupakan *otherwise nibble* dan Δy merupakan *active nibble*, dengan alasan yang sama ditunjukkan bahwa Δz bersifat *otherwise nibble*.

Sifat 5. *Passive nibble* di-XOR-kan dengan *otherwise nibble* (dan sebaliknya) menghasilkan *otherwise nibble*.
Bukti.
Misal terdapat *passive nibble* Δx, *otherwise nibble* Δy, dan Δz yang merupakan hasil XOR antara Δx dan Δy. Δy dapat bersifat *active* atau *passive*. Berdasarkan Sifat 1, Δz bersifat *passive* apabila Δy bersifat *passive*. Apabila Δy bersifat *active*, maka berdasarkan sifat 2, Δz bersifat *active*. Berdasarkan hal tersebut, Δz bersifat *otherwise*. Demikian pula jika Δx merupakan *otherwise nibble* dan Δy merupakan *passive nibble*, dengan cara yang sama terbukti bahwa Δz bersifat *otherwise*.

Sifat 6. *Otherwise nibble* di-XOR-kan dengan *otherwise nibble* menghasilkan *otherwise nibble*.
Bukti.
Misal terdapat Δx dan Δy yang merupakan *otherwise nibble* dan Δz adalah hasil XOR antara kedua *nibble difference* tersebut. Terdapat dua kemungkinan kondisi untuk Δx dan Δy, yakni bersifat *active* atau *passive*. Apabila Δx dan Δy bersifat *passive*, maka berdasarkan sifat 1, Δz bersifat *passive*. Jika Δx dan Δy bersifat *active*, maka Δz bersifat *otherwise* berdasarkan sifat 3. Berdasarkan sifat 2, Δz bersifat *active* apabila Δx bersifat *active* (atau *passive*) dan Δy bersifat *passive* (atau *active*). Oleh karena itu, Δz bersifat *otherwise*.

3. METODOLOGI PENELITIAN

Metode penelitian yang digunakan pada penelitian ini adalah metode eksperimen. Metode eksperimen termasuk dalam metode penelitian dengan menggunakan pendekatan kuantitatif. Metode eksperimen merupakan metode penelitian yang digunakan untuk mencari pengaruh dari suatu variabel tertentu. Metode eksperimen digunakan pada penelitian ini dalam melakukan pencarian karakteristik *impossible differential reduced round* pada algoritme SKINNY menggunakan teknik *miss-in-the-middle*.

Berikut adalah tahapan untuk mencari *impossible differential path* terbaik pada SKINNY dengan mencoba kombinasi satu dan dua *active nibble* pada semua kemungkinan varian.

a. Mengumpulkan referensi dan literatur, mempelajari, dan mengkaji teori-teori dari berbagai sumber seperti paper, buku, dan sebagainya. Kajian kepustakaan meliputi teori-teori yang berkaitan dengan konsep dasar *lightweight block cipher*, algoritme SKINNY, serangan *impossible differential* dan teknik *miss-in-the-middle*, serta materi pendukung lainnya.
b. Melakukan observasi terhadap komponen yang terdapat pada algoritme SKINNY.
c. Membangkitkan *input* dan *output difference*.
d. Mencari *forward differential path* SKINNY.
e. Mencari *backward differential path* r^* *round* SKINNY.
f. Mencari *impossible differential path reduced round* algoritme SKINNY dengan menggunakan teknik *miss-in-the-middle*.
Untuk mencari *impossible differential path*, maka dilakukan pencarian *path* sesuai skenario *forward* dan *backward* dengan skenario yang ditunjukkan pada Tabel 3.

Tabel 3 Skenario *forward* dan *backward differential path* pada pencarian *impossible differential path*

Skenario.	*Forward Differential Path*	*Backward Differential Path*
1.	Satu *active nibble*	Satu *active nibble*
2.	Dua *active nibble*	Satu *active nibble*
3.	Satu *active nibble*	Dua *active nibble*
4.	Dua *active nibble*	Dua *active nibble*

Metode yang digunakan untuk mencari *impossible differential path* r *round* SKINNY adalah *miss-in-the-middle*. Pencarian *impossible differential path* n *round* dilakukan dengan mengkombinasikan *forward differential path* r' *round* dan *backward differential path* r^* *round* dan mencari posisi *nibble* yang memiliki kontradiksi. Kontradiksi yang dimaksudkan adalah adanya perbedaan sifat *nibble difference* pada posisi *nibble* tertentu. Sifat *nibble difference* yang dapat digunakan untuk menentukan kontradiksi adalah *active* dan *passive*. *Otherwise nibble* tidak dapat digunakan untuk menentukan kontradiksi karena tidak diketahui *nibble* tersebut bersifat *active* atau *passive*.

g. Menganalisis karakteristik *impossible differential* terbaik dari *reduced round* algoritme SKINNY. Terdapat tiga jenis analisis, yaitu analisis jumlah dan posisi *active nibble*, analisis kombinasi *forward* dan *backward differential path* dan analisis *penerapan impossible differential path* pada perluasan *round*.
h. Menarik simpulan mengenai penerapan serangan *impossible differential* pada *reduced round* SKINNY.

4. PEMBAHASAN

4.1. Observasi Komponen pada SKINNY

Observasi komponen pada algoritme SKINNY dilakukan untuk mengetahui pengaruh dari setiap komponen yang terdapat pada algoritme SKINNY terhadap *input difference*. Komponen yang dilakukan observasi pada penelitian ini terdiri dari operasi XOR, *SubCells*, *AddConstants*, *AddRoundTweakey*, *ShiftRows*, dan *MixColumns*. Terdapat tiga klasifikasi untuk sifat *nibble*, yaitu *active nibble* (α), *passive nibble* (0), dan *otherwise nibble* (?). *Passive nibble* merupakan *nibble difference* yang bernilai nol ($x = x$'). *Active nibble* merupakan *nibble difference* yang bernilai tidak sama dengan nol ($x \neq x'$). *Otherwise nibble* merupakan *nibble difference* yang dapat bernilai nol (*passive*) atau bernilai selain nol (*active*).

Notasi α menunjukkan sifat *active nibble* dengan $\alpha \in \{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15\}$. Jika ter-dapat dua atau lebih *nibble difference* dengan notasi α, maka *nibble difference* tersebut merupakan *active nibble* dengan nilai tertentu yang berbeda.

1) XOR

Misal terdapat *input nibble difference* Δx_1 dan Δx_2 dengan $\Delta x_1 = x_1 \oplus x_1'$ dan $\Delta x_2 = x_2 \oplus x_2'$. Jika *output* dari Δx_1 dan Δx_2 secara berurutan yaitu Δy_1 dan Δy_2. Nilai Δy merupakan nilai *difference* dari Δy_1 dan Δy_2, maka

$$\Delta y = f(\Delta x_1) \oplus f(\Delta x_2) \quad \text{.....(4.1)}$$

Bukti:

$$\begin{aligned} \Delta y &= \Delta y_1 \oplus \Delta y_2 \\ &= y_1 \oplus y_1' \oplus y_2 \oplus y_2' \\ &= f(x_1) \oplus f(x_1') \oplus f(x_2) \oplus f(x_2') \\ &= f(x_1 \oplus x_1') \oplus f(x_2 \oplus x_2') \\ &= f(\Delta x_1) \oplus f(\Delta x_2) \end{aligned}$$

Persamaan 4.1 menunjukkan bahwa *output nibble difference* dari operasi XOR adalah hasil XOR antara dua *input nibble difference*-nya. *Output nibble difference* operasi XOR untuk semua kemungkinan *input nibble difference* ditunjukkan pada Tabel 4.

Tabel 4. *Output nibble difference* operasi XOR untuk semua kemungkinan *input nibble difference* [11]

No.	Δx_1	Δx_2	Δy	Keterangan
1.	0	0	0	Sifat 1
2.	0	α	α	Sifat 2
3.	α	0	α	Sifat 2
4.	0	?	?	Sifat 5
5.	?	0	?	Sifat 5
6.	α	?	?	Sifat 4
7.	?	α	?	Sifat 4
8.	α	α	?	Sifat 3
9.	?	?	?	Sifat 6

Berdasarkan Tabel 4 dibuktikan bahwa operasi XOR mempengaruhi nilai dari suatu *nibble difference*. Hal ini ditunjukkan dari hasil *output difference* yang memiliki sifat-sifat yang unik terhadap *input difference*-nya.

2) *SubCells*

Pada operasi *SubCells*, setiap nilai pada *nibble* disubstitusikan dengan nilai baru berdasarkan suatu *s-box* SC. Diketahui $SC : F_2^4 \rightarrow F_2^4$ merupakan *s-box* bijektif dari SKINNY. Misal terdapat dua *input s-box* x dan x' dengan *input difference* $\Delta x = x \oplus x'$ dan *output s-box* yaitu $y = SC(x)$ dan $y' = SC(x')$ dengan *output difference* $\Delta y = y \oplus y'$.

Pengaruh *s-box* terhadap sifat dari *nibble difference* dapat dilihat pada sifat 7, sifat 8, dan sifat 9 [11].

Sifat 7. *Passive nibble* pada *input nibble difference* s-box menghasilkan *passive nibble* pada *output nibble difference*.

Sifat 8. *Active nibble* pada *input nibble difference s-box* menghasilkan *active nibble* pada *output nibble difference*. Misal $x = x'$ sehingga $\Delta x \neq 0$, maka $\Delta y \neq 0$.

Sifat 9. *Otherwise nibble* pada *input nibble difference s-box* menghasilkan *otherwise nibble* pada *output nibble difference*.

3) *AddConstants* dan *AddRoundTweakey*

Pada operasi *AddConstants* dan *AddRoundTweakey*, nilai *input difference* tidak berpengaruh. *Output difference* tidak dipengaruhi oleh proses XOR dengan *round constants* maupun *round tweakey* pada *input difference*. Nilai dari *input difference* menjadi *nilai output difference*. Ide dasar dari operasi *AddRoundTweakey* serupa seperti yang ada pada *AddRoundKey* pada algoritme AES. Penjelasan dari *AddConstants* dan *AddRoundTweakey* terdapat pada Sifat 10 berikut.

Sifat 10. *Output difference* dari operasi *AddConstants* dan *AddRoundTweakey* merupakan *input difference*-nya.

Output difference dari operasi *AddConstants* dan *AddRoundTweakey* adalah *input difference*-nya. Walaupun setiap *round* memiliki sifat yang dinamis, namun secara mendasar perubahan dari nilai *round constants* dan *tweakey* tidak mengubah sifat *input difference*.

4) *ShiftRows* dan *InvShiftRows*

ShiftRows dan *InvShiftRows* merupakan permutasi *nibble* dengan menggeser beberapa *nibble* berdasarkan posisi baris *nibble* pada SKINNY. Sifat 11 menunjukkan hasil *output difference* operasi *ShiftRows* terhadap suatu *input difference*.

Sifat 11. *Output difference* dari operasi *ShiftRows* merupakan hasil permutasi *ShiftRows* dari *input difference*-nya.

Output difference dari operasi *ShiftRows* merupakan hasil permutasi *ShiftRows* dari *input difference*-nya, sehingga operasi *ShiftRows* menyebabkan perubahan posisi pada *input difference*.

5) *MixColumns*

Hasil *output* dari operasi *MixColumns* merupakan penggabungan hasil perkalian matriks X dengan masing-masing kolom *state* x_i yang dilakukan secara terpisah. Anak subbagian ini ditunjukkan hasil observasi dari perkalian matriks untuk masing-masing kolom dan *MixColumns* untuk sebuah nibble x_i.

Misal terdapat input perkalian matriks $x = \{x_i, 0 \leq i \leq 3\}$ dan $x' = \{x_i', 0 \leq i \leq 3\}$ dengan *input difference* $\Delta x = x \oplus x' = \{\Delta x_i, 0 \leq i \leq 3\}$ dan *output* perkalian matriks $y = MC(x) = \{y_i, 0 \leq i \leq 3\}$ dan $y^{\wedge\prime} = MC(x') = \{y_i', 0 \leq i \leq 3\}$ dengan *output difference* $\Delta y = y \oplus y' = \{\Delta y_i, 0 \leq i \leq 3\}$. Operasi perkalian matriks X merupakan XOR antara *nibble* pada kolom dari *nibble* yang dapat dinyatakan dengan persamaan berikut:

$$\begin{aligned} y_0 &= x_0 \oplus x_2 \oplus x_3, \\ y_1 &= x_0, \\ y_2 &= x_1 \oplus x_2, \\ y_3 &= x_0 \oplus x_2 \end{aligned}$$

Berdasarkan persamaan tersebut, maka *output difference* dari perkalian matriks terhadap suatu *input difference* adalah

$$\Delta y = y \oplus y' = MC(x) \oplus MC(x') = MC(\Delta x)$$

Bukti:

$$\begin{aligned} \Delta y &= y \oplus y' \\ &= MC(x) \oplus MC(x') \\ &= \begin{pmatrix} x_0 \oplus x_2 \oplus x_3 \\ x_0 \\ x_1 \oplus x_2 \\ x_0 \oplus x_2 \end{pmatrix} \oplus \begin{pmatrix} x_0' \oplus x_2' \oplus x_3' \\ x_0' \\ x_1' \oplus x_2' \\ x_0' \oplus x_2' \end{pmatrix} \\ &= \begin{pmatrix} x_0 \oplus x_2 \oplus x_3 \oplus x_0' \oplus x_2' \oplus x_3' \\ x_0 \oplus x_0' \\ x_1 \oplus x_2 \oplus x_1' \oplus x_2' \\ x_0 \oplus x_2 \oplus x_0' \oplus x_2' \end{pmatrix} \\ &= \begin{pmatrix} (x_0 \oplus x_0') \oplus (x_2 \oplus x_2') \oplus (x_3 \oplus x_3') \\ (x_0 \oplus x_0') \\ (x_1 \oplus x_1') \oplus (x_2 \oplus x_2') \\ (x_0 \oplus x_0') \oplus (x_2 \oplus x_2') \end{pmatrix} \\ &= \begin{pmatrix} \Delta x_0 \oplus \Delta x_2 \oplus \Delta x_3 \\ \Delta x_0 \\ \Delta x_1 \oplus \Delta x_2 \\ \Delta x_0 \oplus \Delta x_2 \end{pmatrix} \\ &= MC(\Delta x) \end{aligned}$$

Terbukti bahwa *output difference* dari perkalian matriks merupakan hasil operasi perkalian matriks terhadap *input difference*-nya.

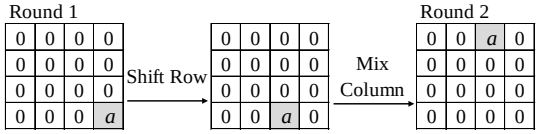
4.2. Pembangkitan *Input* dan *Output Difference*

Terdapat dua jenis *input* dan *output difference* yang digunakan untuk mencari *forward* dan *backward differential path*, yakni satu *active nibble* dan dua *active nibble* berbeda. *String nibble output difference* yang dibangkitkan sama dengan *input difference*. Kondisi satu *active nibble* membangkitkan 16 kemungkinan *string nibble*, sehingga terdapat 16 kemungkinan *input* dan *output difference* untuk satu *active nibble*. Kondisi dua *active nibble* membangkitkan 120 kemungkinan *string nibble*, sehingga terdapat 120 kemungkinan *input* dan *output difference* untuk dua *active nibble*.

4.3. Pencarian *Forward Differential Path*

Input round function pada *round* r yang selanjutnya dinotasikan dengan ΔX^r dengan $X = (x_0, x_1, \dots, x_{15})$ dan $0 \leq r \leq n$. *Round* terpanjang yang diperoleh pada pencarian *forward differential* yang selanjutnya dinotasikan dengan r'. Berikut dijelaskan mengenai ilustrasi pencarian *forward differential path* r' *round* SKINNY. Ilustrasi dilakukan untuk *forward differential path* 6 *round* dengan *input difference* satu *active nibble* pada posisi x_{15}, yakni ΔX^0=000000000000000a.

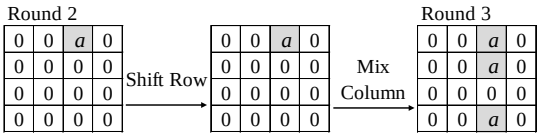
Input difference ΔX^0 merupakan *input difference* untuk operasi *ShiftRows*. *Output difference* dari operasi *ShiftRows round* ke-1 adalah $\Delta X^{0,5}$=00000000000000a0. *Output difference* dari operasi *ShiftRows* menjadi *input difference* untuk operasi *MixColumns*. *Input difference* untuk operasi *MixColumns* adalah $\Delta X^{0,5}$=00000000000000a0. *Output difference* dari operasi *MixColumns round* ke-1 adalah ΔX^1=00a0000000000000. *Output difference* dari operasi *MixColumns* merupakan *output difference* untuk *round* ke-1 dan menjadi *input difference* untuk *round* ke-2.Gambar 4 menunjukkan hasil operasi *input difference* ΔX^0=000000000000000a pada *round* ke-1.



Gambar 4. *Forward differential path round* ke-1 untuk $\Delta X^0 = 000000000000000a$

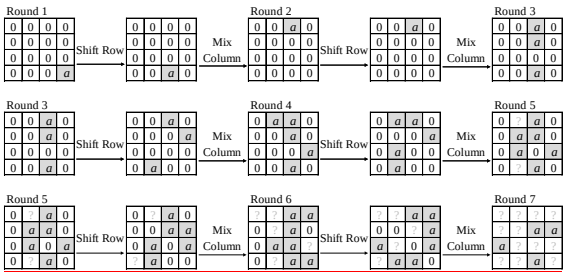
Input difference untuk *round* ke-2 adalah ΔX^1=00a0000000000000. Pada *round* ke-2, operasi yang mempengaruhi *nibble difference* untuk pencarian *forward differential path* sama seperti pada *round* ke-1 yaitu *ShiftRows* dan *MixColumns*. *Output difference* dari operasi *ShiftRows* menjadi *input difference* untuk operasi *MixColumns*. Selanjutnya, pencarian *path* untuk operasi *ShiftRows* dan *MixColumns* di *round* ke-2 dilakukan sama seperti proses pencarian *path round* ke-1. Hasil pencarian *path round* ke-2 pada *forward differential path* 6 *round* SKINNY dapat dilihat pada Gambar 5. *Output*

difference yang dihasilkan pada *round* ke-2 adalah ΔX^2=00a000a0000000a0.



Gambar 5. *Forward differential path round* ke-2 untuk $\Delta X^1 = 00a0000000000000$

Output difference dari *round* ke-2 menjadi *input difference* untuk *round* ke-3. Pencarian *path* di *round* ke-3 hingga *round* ke-6 dilakukan dengan proses yang sama dengan proses pencarian *path* di dua *round* pertama. *Output difference* yang dihasilkan hingga *round* ke-6 adalah ΔX^6=??????aaa?????a?. Gambar 6 menunjukkan *path* hingga *round* ke-6 pada proses pencarian pencarian *forward differential* path 6 *round* SKINNY.

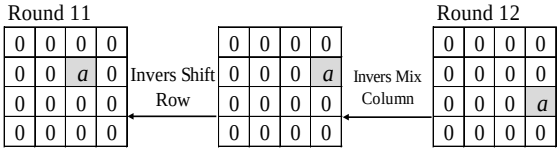


Gambar 6. *Forward differential path round* ke-6 SKINNY

4.4. Pencarian *Backward Differential Path*

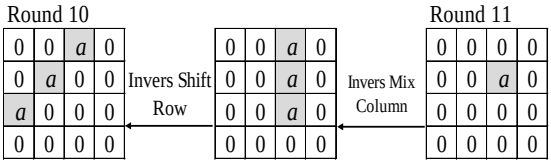
Round terpanjang yang diperoleh pada pencarian *backward differential* yang selanjutnya dinotasikan dengan r^*. Berikut dijelaskan mengenai ilustrasi pencarian *backward differential path* r^* *round* SKINNY. Ilustrasi dilakukan untuk *backward differential* path r^* *round* dengan *output difference* satu *active nibble* pada posisi x_{11}, yakni ΔY^r=00000000000a0000. Misal r yang digunakan adalah 12 dan r^* bernilai 5, maka dicari *backward differential path* 5 *round* mulai *round* ke-12 sampai dengan *round* ke-8. *Input difference* dari operasi *MixColumns round* ke-11 adalah ΔY^r=00000000000a0000.

Input difference dari operasi *MixColumns* menjadi *output difference* untuk operasi *InvShiftRows*. *Input difference* dari operasi *MixColumns round* ke-12 adalah $\Delta Y^{r-0,5}$=0000000a00000000. *Output difference* untuk operasi *InvShiftRows* adalah *input difference* dari operasi *MixColumns*, yakni ΔY^{r-1}=000000a000000000. *Input difference* dari operasi *InvShiftRows round* ke-11 adalah ΔY^{r-1}=000000a000000000. *Input difference* dari operasi *InvShiftRows* merupakan *input difference* untuk *round* ke-11 dan menjadi *output difference* untuk *round* ke-10. Gambar 7 menunjukkan hasil operasi *output difference* pada *round* ke-11.



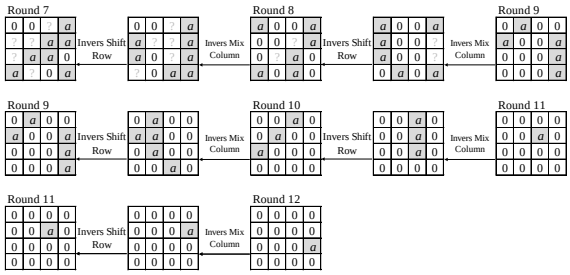
Gambar 7. *Backward differential path round* ke-r untuk ΔY^r = 00000000000a0000

Output difference untuk *round* ke-10 adalah ΔY^{r-1}=000000a000 000000. Selanjutnya, proses pencarian *path round* ke-10 dilakukan seperti pada *round* ke-11. Penggunaan metode yang sama dalam proses pencarian menghasilkan *input difference* ΔY^{r-2}=00a00a00a0000000 pada *round* ke-10. *Input difference* dari *round* ke-10 kemudian menjadi *output difference* pada *round* ke-9. *Path round* ke-11 pada *backward differential path* 5 *round* SKINNY dapat dilihat pada Gambar 8.



Gambar 8. *Backward differential path round* ke $r-1$ untuk ΔY^{r-1} = 000000a000000000

Pencarian *path* di *round* ke-10 hingga *round* ke-8 dilakukan menggunakan proses yang sama seperti proses pencarian *path* pada *round* ke-12 dan *round* ke-11. Pada *round* ke-8, didapatkan hasil *input difference* ΔY^{r-5}=00?a??aa?aa0a?0a. *Input difference* tersebut merupakan *input difference backward differential path* 5 *round* SKINNY dengan *output difference* ΔY^r=00000000000a0000. Gambar 9 menunjukkan hasil *backward differential path* pada *round* ke-8.

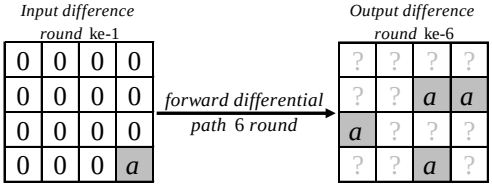


Gambar 9. *Backward differential path round* ke $r-5$

4.5. Pencarian *Impossible Differential Path*

Berikut ini dijelaskan contoh ilustrasi pencarian *impossible differential path* pada 11 *round* SKINNY. Kombinasi yang dapat digunakan untuk mendapatkan *impossible differential path* 11 *round* adalah kombinasi *forward differential path* 6 *round* dan *backward differential path* 5 *round* atau sebaliknya. Karakteristik *impossible differential* yang digunakan yaitu 000000000000000α $\nrightarrow$ 00000000000a0000. Misal terdapat *input difference* ΔX^0=000000000000000α pada *forward differential path* 6 *round* SKINNY. Subbagian 4.3 telah

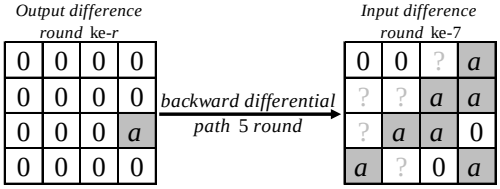
dijelaskan bahwa *input difference* tersebut menghasilkan *output difference* pada *round* ke-6 adalah ΔY^6=??????$\alpha\alpha\alpha$?????α?.



Gambar 10. *Forward differential path* 6 *round* untuk $\Delta X^0 = 0000000000000000\alpha$

Output difference round ke-6 pada *forward differential path* tersebut memiliki *otherwise nibble* pada posisi x_0, x_1, x_2, x_3, x_4, x_5, x_9, x_{10}, x_{11}, x_{12}, x_{13}, dan x_{15}. *Active nibble* terdapat pada posisi x_6, x_7, x_8, dan x_{14}. Proses pencarian *impossible differential path* dilakukan pencarian *backward differential path* dengan posisi *nibble* yang membentuk kontradiksi pada posisi x_6, x_7, x_8, dan x_{14} karena *active nibble* dapat digunakan untuk menentukan kontradiksi.

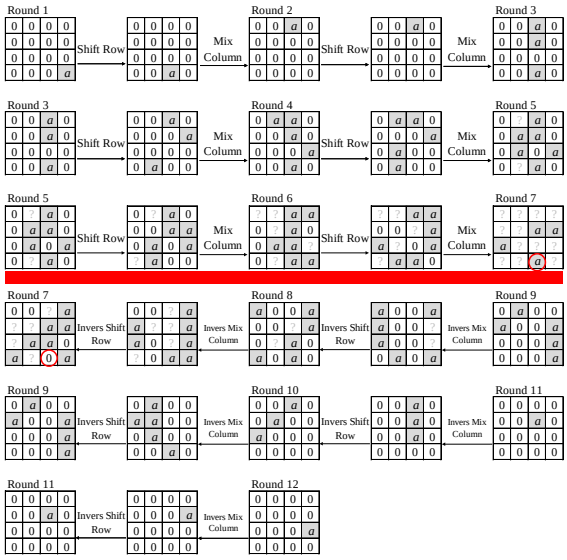
Pencarian *backward differential path* 5 *round* algoritme SKINNY untuk *output difference* ΔY^r=00000000000a0000 telah dijelaskan pada subbagian 4.4. Hasil *input difference round* ke-7 untuk *output difference* tersebut adalah ΔX^7=00?a??aa?aa0a?0a.



Gambar 11. *Backward differential path* 5 *round* untuk $\Delta Y^r = 00000000000\alpha0000$

Output difference tersebut memiliki hasil *input difference* yang bersifat *passive* pada posisi *nibble* x_0, x_1, x_{11}, dan x_{14}. *Backward differential path* dengan *output difference* ΔY^r=00000000000a0000 dapat dikombinasikan dengan *forward differential path* dengan *input difference* ΔX^0=000000000000000a. Kontradiksi pada kombinasi *forward* dan *backward differential path* ini terdapat pada posisi *nibble* x_{14}. Hal ini menunjukkan bahwa terdapat *impossible differential path* 11 *round* SKINNY dengan *input difference* ΔX^0=000000000000000a dan *output difference* ΔY^r=00000000000a0000 untuk 11 *round* SKINNY. Ilustrasi pencarian *impossible differential path* 11 *round* SKINNY tersebut ditunjukkan pada Gambar 12.

Pada penelitian ini dilakukan pencarian kontradiksi pada empat kombinasi *forward* dan *backward differential path* untuk menentukan *impossible differential path*. Jumlah *round* terpanjang pada kombinasi satu *active nibble* pada *forward* dan *backward differential path* yaitu 11 *round*.



Gambar 12. *Impossible differential path* 11 *round* SKINNY

4.6. Analisis Penerapan *Impossible Differential Path* Pada Perluasan *Round*

Analisis ini bertujuan untuk mendapatkan *impossible differential path* yang digunakan dalam perluasan karakteristik *impossible differential* 16 *round*. Perluasan karakteristik dilakukan dengan menambah beberapa *round* di atas *impossible differential path* berdasarkan *input difference*. Ketika *active nibble* diperluas ke atas maka menghasilkan penambahan *active nibble*. Hal ini terjadi karena operasi *invers MixColumns* sehingga menyebabkan penyebaran *active nibble* dan menghasilkan penambahan *active nibble*.

Pada penelitian ini digunakan contoh penerapan serangan *impossible differential path* skenario 1, karena untuk perluasan ke bawah skenario 1 lebih banyak menghasilkan penambahan *active nibble*., semakin banyak penambahan *active nibble* maka semakin menambah jumlah *active nibble* pada *output* akhir. Semakin banyak jumlah *active nibble* pada *output* akhir maka semakin besar ditemukanya pasangan *plaintext* dan *ciphertext* yang memenuhi karakteristik *impossible differential*. Contoh kasus *impossible differential path* 11 *round* SKINNY yang digunakan pada skenario 1 adalah $000000000000\alpha000 \nrightarrow 00000000\alpha0000000$.

Proses yang dilakukan pada serangan *impossible differential* 11 *round* SKINNY dengan skenario 1 adalah sebagai berikut:

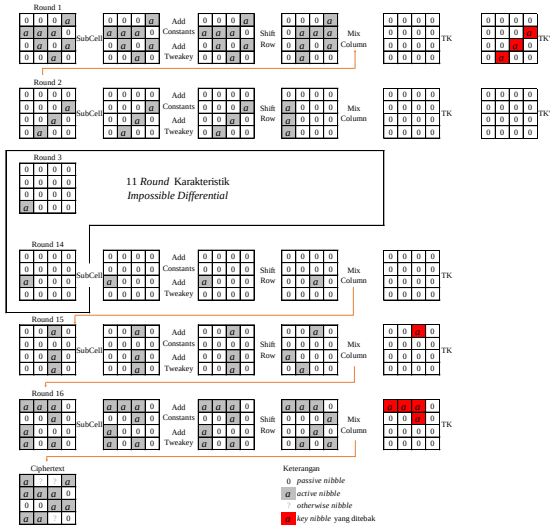
a. Memperluas *impossible differential path* 11 *round* SKINNY.

   Serangan *impossible differential reduced round* pada SKINNY dilakukan dengan memperluas *impossible differential path* 11 *round*. Perluasan 16 *round* untuk skenario 1 dapat dilakukan dengan menambahkan *round* di atas dan di bawah. Perluasan pada contoh kasus ini dilakukan dengan dua *round* ke atas dan tiga

round ke bawah. Gambar 13 menunjukkan perluasan 11 *round* pada *impossible differential path* $000000000000a000 \nrightarrow 00000000a0000000$. Hasil dari perluasan *round* yaitu terdapat beberapa *nibble subtweakey* yang ditebak, terdiri dari 3 *nibble* pada dua *round* pertama dan 5 *nibble* pada tiga *round* terakhir. Jadi, seluruhnya terdapat 8 *nibble subtweakey* yang ditebak.

b. Fase pengumpulan data.

Pada fase ini dilakukan pengumpulan data yang dibutuhkan untuk melakukan proses *key recovery*. Pertama-tama membentuk suatu struktur 2^{3c} *plaintext* pada *input round* 2. *Plaintext* memiliki nilai tetap pada *nibble* $x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_8, x_9, x_{11}, x_{12}, x_{14}, \quad x_{15}$ dan tiga posisi lainnya dapat berisi semua kemungkinan nilai. Setiap struktur dapat membentuk 2^{6c-1} pasangan *plaintext*. Untuk melakukan serangan, diperlukan 2^n struktur sehingga terdapat 2^{n+6c-1} pasangan *plaintext* yang dapat dibentuk dari 2^{n+3c} *plaintext*.



Gambar 13 Perluasan 11 *round impossible differential path* pada SKINNY skenario 1

c. Fase *Recovery* Kunci.

Fase *recovery* kunci dilakukan dengan langkah-langkah sebagai berikut:

1) Melakukan enkripsi pada sisa pasangan *plaintext* sebanyak 15 *round* untuk mendapatkan ΔX_{16}. Selanjutnya diambil pasangan *plaintext* yang memiliki *passive nibble* sejumlah 9 *nibble* sebelum operasi *MixColumns* terakhir yang terdapat pada posisi $x_3, x_4, x_5, x_6, x_8, x_9, x_{11}, \ x_{12}$ dan x_{14}. Pasangan *plaintext* yang sesuai dengan kondisi tersebut ada sebanyak $2^{n+6c-1-9c} = 2^{n-3c-1}$.
2) Setiap pasangan pasangan *plaintext* tersebut dapat menghasilkan semua nilai *subtweakey* sebanyak 8 *nibble* yang mengarah ke karakteristik *impossible differential* dengan menebak 5 *nibble* pada *internal state*, yang terdiri dari 1 *nibble difference* setelah operasi *MixColumns* pada *round* 3 dan 4 *nibble difference* sebelum operasi *MixColumns* pada *round* 14 dan *round* 15. Jadi diperoleh $2^{n-3c-1+5c} = 2^{n+2c-1}$ saran kunci yang salah untuk 8 *nibble subtweakey*.
3) Kunci-kunci yang ditebak dan memenuhi kondisi *impossible differential* merupakan kunci yang salah untuk dieliminasi. Selain itu, kunci-kunci yang tidak memenuhi kondisi *impossible differential* namun merupakan kunci yang salah juga dieliminasi. Jumlah sisa yang diharapkan dari 8 *nibble subtweakey* yang salah adalah $N = 2^{8c} \times (1 - 2^{-8c})^{2^{n+2c-1}}$. Untuk mendapatkan $N < 1$ dipilih $n = 29{,}5$, sehingga $N = 2^{32} \cdot 2^{-32{,}6} = 2^{-0{,}6}$.
Jadi, kompleksitas data untuk serangan *impossible differential* 11 *round* pada SKINNY dengan skenario 1 adalah 2^{n+3c} *plaintext*.

Pada penelitian ini dilakukan percobaan penebakan kunci pada ke-4 skenario berdasarkan contoh kasus menggunakan salah satu karakteristik yang ditemukan Beirle, *et al*., yaitu (000000000000a000↛00000000a0000000) dengan perluasan round ke atas dan ke bawah. Dari hasil percobaan, jumlah kunci yang berhasil ditebak dengan kompleksitas data yang masih lebih kecil dari batas ambang *brute force attack* (2^{64}) dapat dilihat pada Tabel 5 yang bertanda kuning.. Terlihat skenario 1 yang mencapai round terpanjang yatiu 16. Jika dilanjutkan hingga 16-round pada skenario 2, 3 dan 4, maka kompleksitas data bertambah signifikan melebihi 2^{64} seperti terlihat pada Tabel 5.

Tabel 5. Rekapitulasi skenario dari contoh kasus serangan *impossible differential reduced round* SKINNY

Skenario	Kompleksitas Data	Jumlah kunci yang ditebak	Panjang *Round*
1	$2^{41,5}$	32 bit	16
2	$2^{40,8}$	20 bit	14
	$2^{77,6}$	72 bit	16
3	$2^{37,6}$	36 bit	15
	$2^{81,7}$	76 bit	16
4	$2^{37,06}$	24 bit	13
	$2^{138,5}$	132 bit	16

Berdasarkan rekapitulasi dari contoh kasus pada empat skenario diperoleh bahwa skenario 1 yang mencapai round terpanjang dengan kompleksitas $2^{41,5}$ sebagai skenario terbaik. Hasil ini mengkonfirmasi skenario serangan yang dilakukan oleh Beierle *et al*. (2016) menggunakan salah satu dari 16 karakteristik yang ditemukan Beirle *et al*., dengan kompleksitas data $2^{41,5}$. Terbukti skenario serangan *impossible differential reduced round* SKINNY milik Beierle *et al.* (2016) merupakan skenario serangan

impossible differential terbaik. Tabel 6 menunjukkan bit subtweakey yang bisa ditebak berdasarkan jumlah kombinasi perluasan *round* ke atas dan ke bawah pada skenario 1.

Tabel 6. Bit *subtweakey* yang bisa ditebak berdasarkan jumlah perluasan *round*

Perluasan		Bawah			
		0	1	2	3
Atas	0	0	0	4	20
	1	0	0	4	20
	2	12	12	16	32
	3	40	40	44	60

Berdasarkan Tabel 6, semakin bertambah jumlah perluasan *round* maka semakin banyak bit *subtweakey* yang bisa ditebak, namun kompleksitas data yang diperlukan juga semakin besar. Hal ini juga terbukti berlaku untuk tiga skenario lainnya yang hasilnya ditunjukkan pada Tabel 5.

5. SIMPULAN DAN SARAN

Berdasarkan hasil penelitian dan analisis yang telah dilakukan, skenario serangan *impossible differential reduced round* algoritme SKINNY milik Beierle *et al.* (2016) atau skenario 1 merupakan skenario serangan *impossible differential* terbaik dikarenakan mencakup jumlah *round* yang lebih panjang dengan nilai kompleksitas yang masih lebih kecil dari batas ambang *brute force attack*. Skenario 1 membutuhkan kompleksitas sebanyak $2^{41,5}$ *plaintext* untuk menebak 32 bit *subtweakey* dengan perluasan panjang *round* sebanyak 16 *round*, sedangkan tiga skenario lain membutuhkan kompleksitas yang lebih besar dari skenario 1, yaitu $2^{77,6}$, $2^{81,7}$, dan $2^{138,5}$ untuk perluasan *round* hingga 16 *round*. Oleh karena itu dapat disimpulkan bahwa semakin bertambah jumlah perluasan *round* maka semakin banyak bit *subtweakey* yang bisa ditebak, namun kompleksitas data yang diperlukan juga semakin besar.

Saran untuk pengembangan selanjutnya adalah perlu dilakukan pencarian *impossible differential path* pada SKINNY dengan metode selain *miss-in-the-middle* dan mencari kompleksitas data terbaik untuk skenario 2, skenario 3, dan skenario 4.

Referensi

[1] Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., dan Sim, S. M., The SKINNY family of block ciphers and its low-latency variant MANTIS. In *Annual Cryptology Conference* (pp. 123-153). Springer, Berlin, Heidelberg, 2016

[2] Liskov, M., Rivest, R. L., dan Wagner, D., Tweakable block ciphers. In *Annual International Cryptology Conference* (pp. 31-46). Springer, Berlin, Heidelberg, 2002

[3] Chen, Z., dan Wang, X. Y., Impossible differential cryptanalysis of midori. In *Mechatronics and Automation Engineering: Proceedings of the International Conference on Mechatronics and Automation Engineering* (ICMAE2016) (pp. 221-229), 2017

[4] Chen, J., Futa, Y., Miyaji, A., dan Su, C., Impossible differential cryptanalysis of LBlock with concrete investigation of key scheduling algorithm. *IACR Cryptology ePrint Archive*, 2014, 272, 2014

[5] Katagi, M. & Moriai, S., Lightweight cryptography for the Internet of Things. Sony Corporation, pp.7–10, 2008

[6] Katz, J., Menezes, A. J., Van Oorschot, P. C., dan Vanstone, S. A., *Handbook of applied cryptography*. CRC press, 1996

[7] Amarullah, A. H., *Pencarian Karakteristik Impossible Differential 8 round pada Algoritme MIBS-64 Menggunakan Teknik Miss-In-The-Middle dengan Pendekatan Information Feedback*. Sekolah Tinggi Sandi Negara, 2017

[8] J. Jean, I. Nikolić, dan T. Peyrin, "Tweaks and Keys for Block Ciphers: The TWEAKEY Framework," hal. 274–288, 2014.

[9] Boura, Christina; Plasencia, Maria Naya; Suder, V., Scrutinizing and Improving Impossible Differential Attacks : Applications to CLEFIA , Camellia , LBlock and Simon (Full Version), 2014

[10] Rahmat, I. S., *Impossible Differential Attack pada LBlock-s*. Sekolah Tinggi Sandi Negara, 2016

[11] Manalu, E. B. S., *Impossible Differential Cryptanalysis pada 10 round Algoritme Midori-64*. Sekolah Tinggi Sandi Negara, 2018

Penilaian Tingkat Kapabilitas Tata Kelola Teknologi Informasi (TI) Berdasarkan Ruang Lingkup Sistem Pemerintahan Berbasis Elektronik (SPBE) Menggunakan *Framework* COBIT 5 (Studi Kasus: Bagian ABC Instansi XYZ)

Alya Zulfatunajja[1)] dan Obrina Candra Briliyant[2)]
(1) Politeknik Siber dan Sandi Negara / alya.zulfatunajja@poltekssn.ac.id
(2) Politeknik Siber dan Sandi Negara / obrina@poltekssn.ac.id

Abstrak

Peraturan Presiden Nomor 95 Tahun 2018 tentang SPBE menyatakan bahwa XYZ merupakan salah satu instansi pemerintah yang memiliki urgensi untuk melakukan evaluasi pemanfaatan TI dalam menunjang proses bisnisnya. Berdasarkan SOTK XYZ, pengelolaan TI pada XYZ berpusat pada bagian ABC. Pada penelitian ini, evaluasi dilakukan dengan cara menilai tingkat kapabilitas tata kelola TI berdasarkan ruang lingkup SPBE menggunakan metode PAM framework COBIT 5. Penelitian ini terdiri dari enam langkah yaitu pengumpulan data, pemetaan ruang lingkup, penilaian tingkat kapabilitas, analisis kesenjangan, validasi, dan pemberian rekomendasi. Hasil penilaian tingkat kapabilitas tata kelola TI ABC berdasarkan 18 aspek ruang lingkup SPBE yang dipetakan pada 16 IT process COBIT 5 yaitu 14 IT process berada pada kapabilitas tingkat 1 (performed) dan 2 IT process berada pada kapabilitas tingkat 2 (managed) sehingga terdapat 14 IT process yang tidak memenuhi target kapabilitas yang diharapkan. Rekomendasi perbaikan akan dibuat berdasarkan proses atribut COBIT 5 untuk setiap IT process dan prioritas rekomendasi akan dilakukan melalui identifikasi pokok permasalahan pengelolaan TI berdasarkan Laporan Kinerja ABC tahun 2018 yaitu pada proses APO02, APO03, APO07, DSS04, APO12 dan DSS02.

Kata Kunci: *COBIT 5* (1), *kapabilitas* (2), *SPBE* (3), *tata kelola TI* (4).

1. PENDAHULUAN

Perkembangan teknologi informasi (TI) yang semakin pesat dari waktu ke waktu telah memberikan dampak positif, salah satunya pada sektor instansi pemerintahan Indonesia melalui implementasi *e-Goverment.* Dalam upaya men-dukung implementasi *e-Government,* Pemerintah Indonesia mengeluarkan Peraturan Presiden Nomor 95 Tahun 2018 tentang Sistem Pemerintahan Berbasis Elektronik (SPBE). Menurut Peraturan Presiden Nomor 95 Tahun 2018, SPBE adalah penyelenggaraan pemerintahan yang memanfaatkan teknologi informasi dan komunikasi untuk memberikan layanan kepada pengguna SPBE [1].

Pemanfaatan TI secara optimal menjadi hal yang penting dalam pelaksanaan SPBE di instansi pemerintahan. Hal tersebut selaras dengan pernyataan Weber [2] yang menyebutkan bahwa keberhasilan organisasi akan sangat dipengaruhi oleh kemampuan dalam memanfaatkan TI secara optimal. Kehadiran TI memberikan peluang bagi organisasi dalam meningkatkan efisiensi, namun secara bersamaan kesempatan juga memberikan risiko [3]. Penerapan TI dalam suatu organisasi juga harus seimbang dengan investasi yang dikeluarkan [4]. Ketergantungan organisasi terhadap TI, adanya risiko terkait TI, bahkan jumlah investasi yang besar pada TI menimbulkan kebutuhan akan *Information Technology Governance* atau tata kelola teknologi informasi [3]. Tata kelola TI yang kurang baik akan menyebabkan beberapa masalah seperti terjadinya kehilangan, perusakan, pencurian, dan penyadapan data penting perusahaan yang akan berdampak buruk pada kerahasiaan, keutuhan, dan ketersediaan informasi dalam proses bisnis maupun layanan yang diberikan oleh instansi pemerintahan [5].

Berdasarkan Peraturan Presiden Nomor 95 Tahun 2018 tentang Sistem Pemerintahan Berbasis Elektronik (SPBE) dinyatakan bahwa pelaksanaan SPBE di suatu instansi pemerintahan harus dipantau dan dievaluasi untuk mengukur dan meningkatkan kualitas implementasi SPBE. Berdasarkan peraturan tersebut, maka semua instansi pemerintahan yang memanfaatkan penggunaan TI dalam proses bisnisnya memiliki kewajiban untuk melakukan evaluasi pemanfaatan TI dalam pelaksanaan SPBE.

Salah satu kementerian di Indonesia yang memiliki urgensi untuk melakukan evaluasi pelaksanaan SPBE ialah XYZ. Berdasarkan Peraturan Menteri XYZ Republik Indonesia Nomor 2 Tahun 2016 tentang Organisasi dan Tata Kerja XYZ, pengelolaan teknologi informasi dan komunikasi pada XYZ berpusat pada bagian ABC. Bagian ABC mempunyai tugas untuk melaksanakan penyusunan kebijakan teknis, pelaksanaan, pemantauan, evaluasi,

serta pelaporan dalam pengembangan dan pengelolaan sistem keamanan informasi dan persandian, teknologi informasi dan komunikasi, dan sistem komunikasi berita pada XYZ dan Perwakilan RI [6]. Menurut Kepala Bidang Perencanaan dan Tata Kelola Teknologi Informasi dan Komunikasi, tata kelola TI yang tidak baik di bagian ABC dapat menyebabkan terganggunya proses bisnis yang ada dan tentunya juga akan berdampak pada masalah keamanan informasi yang disediakan oleh bagian ABC [7].

Dalam melakukan evaluasi tata kelola teknologi informasi, terdapat beberapa kerangka kerja yang dapat dijadikan pedoman secara internasional, seperti PRINCE2, ITIL, ISO dan COBIT. Dari semua kerangka kerja yang ada, *Control Objective Information and Related Technology* (COBIT) yang dikeluarkan oleh *Information Systems Audit and Control Association* (ISACA) merupakan kerangka kerja yang populer dalam perencanaan audit TI dan *leading framework* untuk tata kelola teknologi informasi [8]. Kerangka kerja COBIT digunakan sebagai panduan suatu organisasi dalam melakukan pengendalian akan pengelolaan TI secara optimal dalam proses bisnisnya untuk mencapai tujuan organisasi dengan menyediakan *best practice* sesuai dengan fokus *output* yang diinginkan.

Oleh karena itu, dengan adanya urgensi evaluasi pemanfaatan TI dalam pelaksanaan SPBE berdasarkan Peraturan Presiden Nomor 95 Tahun 2018 dan terdapat sebuah *framework* COBIT 5 yang dapat digunakan untuk evaluasi tata kelola TI, maka pada penelitian ini dilakukan penilaian tingkat kapabilitas tata kelola TI berdasarkan ruang lingkup SPBE yang telah didefinisikan pada Peraturan Presiden Nomor 95 Tahun 2018 menggunakan metode *Process Assesment Model* (PAM) pada *framework* COBIT 5 (pada bagian tulisan selanjutnya disingkat PAM). Penelitian ini bertujuan untuk mengetahui sejauh mana tata kelola TI di bagian ABC instansi XYZ telah sesuai dengan tujuan organisasi untuk memberikan layanan pemerintah yang efektif, efisien, dan menjamin keamanan informasi yang diberikan dalam pelaksanaan SPBE.

2. LANDASAN TEORI

2.1. Tata Kelola Teknologi Informasi

Tata kelola mengenai TI telah didefinisikan oleh beberapa ahli, seperti berikut:

a. Menurut Oltsik [9] tata kelola TI adalah kumpulan kebijakan, proses, aktivitas, dan prosedur untuk mendukung pengoperasian TI agar hasil sejalan dengan strategi yang ditetapkan. Tata kelola TI yang baik harus berkualitas, *well-defined*, dan bersifat *repeatable processes* yang terukur.
b. Menurut Gondodiyoto [10] tata kelola TI merupakan salah satu bagian terpenting dari penerapan *good governance*. Tata kelola TI memadukan proses perencanaan, pengelolaan, penerapan, pelaksanaan, dan pengawasan untuk memberikan jaminan bahwa TI mendukung pencapaian tujuan.
c. ITGI [11] menyatakan bahwa tata kelola TI diperlukan untuk memastikan TI yang digunakan perusahaan dapat membantu perusahaan mencapai strategi dan objektif perusahaan.

Menurut ITGI [11], TI sangat penting dalam mendukung dan mencapai tujuan perusahaan dalam menjalankan proses bisnis. Tata kelola TI yang tidak baik akan menimbulkan permasalah yang harus dihadapi oleh organisasi, seperti [12]:

a. Efisiensi dan efektivitas proses bisnis tidak tercapai;
b. Kualitas yang lebih rendah dari yang diharapkan;
c. Kegagalan untuk menciptakan inovasi;
d. Kerugian bisnis, berkurangnya reputasi, dan melemahkan posisi kompetisi.

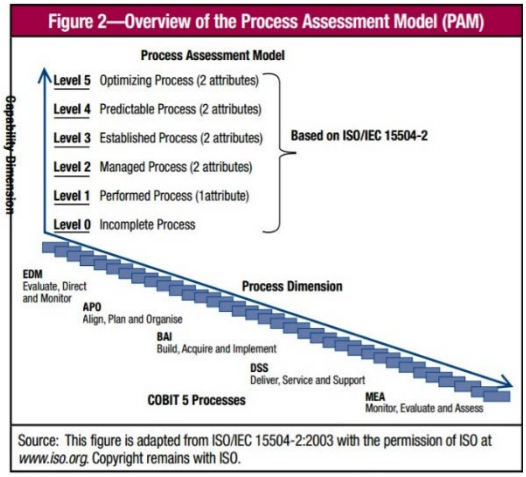
2.2. Sistem Pemerintahan Berbasis Elektronik (SPBE)

Presiden Republik Indonesia berupaya mewujudkan tata kelola pemerintahan yang bersih, efektif, transparan, dan akuntabel serta pelayanan publik yang berkualitas dan terpercaya melalui Sistem Pemerintahan Berbasis Elektronik yang selanjutnya disebut SPBE. Oleh karena itu, dalam upaya meningkatkan keterpaduan dan efisiensi SPBE melalui tata kelola, manajemen, serta audit SPBE, Presiden Republik Indonesia mengeluarkan Peraturan Presiden Republik Indonesia Nomor 95 Tahun 2018 Tentang Sistem Pemerintahan Berbasis Elektronik. SPBE adalah penyelenggaraan pemerintahan dengan memanfaatkan teknologi informasi dan komunikasi untuk memberikan layanan kepada pengguna SPBE. Pelaksanaan SPBE di Instansi Pusat dan Pemerintahan Daerah ditinjau dari tiga aspek yaitu tata kelola, manajemen, dan audit SPBE.

2.3. *Control Objective for Information and Related Technology* (COBIT 5) – *Process Assesment Model* (PAM)

PAM merupakan sebuah model untuk proses kapabilitas yang terdiri dari 2 dimensi. Overview dari PAM diberikan pada Gambar 1. Dimensi pertama merupakan sebuah dimensi proses yang mana proses didefinisikan dan diklasifikasikan menggunakan *Process Reference Model* (PRM) yang terdiri dari lima domain dan terbagi menjadi 37 proses.

Dimensi kedua merupakan dimensi kapabilitas yang merupakan kumpulan dari proses atribut yang dikelompokkan dalan suatu level tertentu, seperti pada Gambar 2.



Gambar 1. COBIT-5 PAM [13]

Figure 4—Capability Levels and Process Attributes

Process Attribute ID	Capability Levels and Process Attributes
	Level 0: Incomplete process
	Level 1: Performed process
PA 1.1	Process performance
	Level 2: Managed process
PA 2.1	Performance management
PA 2.2	Work product management
	Level 3: Established process
PA 3.1	Process definition
PA 3.2	Process deployment
	Level 4: Predictable process
PA 4.1	Process measurement
PA 4.2	Process control
	Level 5: Optimizing process
PA 5.1	Process innovation
PA 5.2	Process optimization

Source: This figure is adapted from ISO/IEC 15504-2:2003 with the permission of ISO at www.iso.org. Copyright remains with ISO.

Gambar 2. Dimensi Kapabilitas dan Proses Atribut [13]

Penentuan level pada dimensi kapabilitas dilakukan dengan suatu indikator penilaian. Indikator penilaian ini digunakan untuk mengetahui sampai sejauh mana pencapaian dari suatu proses atribut. Setiap proses atribut pada dimensi proses PAM akan diukur pencapaiannya melalui suatu skala tertentu yang didefinisikan pada Tabel 1.

Menurut ISACA, suatu proses cukup meraih kategori *Largely Achieved* (L) untuk dapat dinyatakan bahwa proses mencapai level tertentu, namun suatu proses harus meraih kategori *Fully Achieved* (F) untuk dapat melanjutkan ke level kapabilitas berikutnya [7].

Tabel 1. *Rating Scale* [13]

Singkatan	Presentase Pencapaian	Deskripsi
N – Not Achieved	0 – 15%	Pencapaian proses tidak tercapai, pencapaian hanya mempunyai sedikit bukti atau bahkan tidak ada bukti sama sekali.
P – Partially Achieved	15 – 50%	Pencapaian didapatkan setengah terdapat beberapa bukti penggunaan atribut yang terdefinisi. Beberapa aspek dari pencapaian atribut bisa jadi tak terkira.
L – Largely Achieved	50 – 85%	Terdapat bukti penggunaan pendekatan yang sistematis serta pencapaian yang signifikan dari atribut yang terdapat pada proses yang dinilai. Menemukan beberapa kelemahan yang terdapat pada atribut pada saat proses dinilai.
F – Fully Achieved	85 – 100%	Terdapat bukti yang lengkap, pendekatan yang sistematis serta pencapaian yang signifikan dari atribut pada proses yang dinilai. Tidak menemukan kelemahan yang signifikan pada atribut saat proses dinilai.

2.4. Analisis Kesenjangan

Analisis kesenjangan adalah teknik yang digunakan untuk membandingkan dua hal yang dilakukan saat ini dan yang seharusnya dilakukan [14] yang bertujuan untuk melakukan evaluasi kebutuhan pengguna terhadap suatu sistem dan mengidentifikasi apakah kebutuhan tersebut terpenuhi atau tidak oleh sistem [15].

Pada penelitian ini, analisis kesenjangan dilakukan untuk mencari perbedaan antara level tingkat kapabilitas tata kelola TI bagian ABC yang diharapkan dengan level tingkat kapabilitas berdasarkan hasil penilaian menggunakan metode PAM. Hasil dari analisis kesenjangan akan digunakan sebagai pedoman dalam melakukan identifikasi perbaikan untuk peningkatan kapabilitas tata kelola TI bagian ABC berdasarkan proses atribut kerangka kerja COBIT 5 [7].

3. METODOLOGI PENELITIAN

Penelitian ini menggunakan metode penelitian kualitatif dengan pendekatan deskriptif. Penelitian kualitatif melibatkan upaya-upaya penting seperti mengajukan pertanyaan dan prosedur, mengumpulkan data yang spesifik, menganalisis data secara induktif, dan menafsirkan makna data, sedangkan pendekatan deskriptif yaitu penelitian yang memiliki tujuan mendeskripsikan atau menjelaskan sesuatu hal apa adanya [16].

Pada penelitian ini, hal yang diamati adalah pelaksanaan tata kelola TI bagian ABC di XYZ, sehingga penelitian akan berfokus pada pencarian data mengenai sudah sampai sejauh mana penerapan tata kelola TI berhasil mencapai tujuan organisasi melalui penilaian tingkat kapabilitas menggunakan metode PAM berdasarkan ruang lingkup SPBE yang telah didefinisikan pada Peraturan Presiden Nomor 95 Tahun 2018. Tahapan penelitian ini ialah sebagai berikut:

a. Pengumpulan data

Pengumpulan data dibagi menjadi dua yaitu pengumpulan data primer melalui wawancara

kepada narasumber untuk mengetahui sampai sejauh mana penerapan tata kelola TI di bagian ABC dan observasi untuk mengetahui kondisi nyata penerapan tata kelola TI di bagian ABC berdasarkan hasil wawancara dan mencocokkan antara teori yang berlaku dengan kejadian nyata yang terjadi di lapangan untuk menentukan kebutuhan organisasi berkaitan dengan penerapan tata kelola TI. Pengumpulan data sekunder dilakukan melalui melalui telaah kepustakaan. Telaah kepustakaan dilakukan pada segala peraturan, kebijakan, standar, keputusan, maupun dokumen yang berkaitan dengan penerapan tata kelola TI bagian ABC di XYZ.

b. Pemetaan aspek tata kelola TI berdasarkan ruang lingkup SPBE ke dalam ruang lingkup *framework* COBIT 5.

Tahap ini akan memetakan apa saja aspek penerapan tata kelola TI bagian ABC di XYZ berdasarkan ruang lingkup SPBE (Tata Kelola dan Manajemen) pada Peraturan Presiden Nomor 95 Tahun 2018 ke dalam *framework* COBIT 5.

Pemetaan aspek penerapan tata kelola TI di bagian ABC ini akan dilakukan dengan memperhatikan persamaan definisi maupun tujuan aspek dalam SPBE dengan proses pada *framework* COBIT 5. Pemetaan ini merupakan hasil diskusi dengan Kepala Bidang Penyiapan Perumusan Kebijakan Penerapan Sistem Pemerintahan Berbasis Elektronik (SPBE) yaitu Bapak Ugi Cahyo Setiono dan *expert judgment* dari Bapak Dicky Indra Prasetya selaku *Government Relation and Advocacy Director* ISACA Indonesia.

c. Penilaian tingkat kapabilitas tata kelola TI menggunakan PAM

Aspek penerapan tata kelola TI berdasarkan ruang lingkup SPBE yang telah dipetakan ke dalam ruang lingkup COBIT 5 akan berfungsi sebagai IT *process framework COBIT* 5. IT *process* akan dinilai untuk menentukan level kapabilitas penerapan tata kelola TI menggunakan metode PAM.

d. Analisis kesenjangan hasil penilaian tingkat kapabilitas terhadap target pencapaian tata kelola TI

Pada penelitian ini, analisis kesenjangan dilakukan untuk mencari perbedaan antara level tingkat kapabilitas tata kelola TI bagian ABC yang diharapkan dengan level tingkat kapabilitas berdasarkan hasil penilaian menggunakan PAM.

e. Validasi

Validasi akan dilakukan pada hasil penilaian kapabilitas dan hasil gap *analysis* menggunakan *member checking*.

f. Rekomendasi

Hasil penelitian yang sudah tervalidasi akan dijadikan sebagai informasi pencapaian tata kelola TI yang sudah dilakukan dan dijadikan pedoman dalam memberikan rekomendasi terhadap pelaksanaan tata kelola TI bagian ABC di XYZ. Rekomendasi akan dibuat berdasarkan proses atribut metode PAM [7] dan pemrioritasan rekomendasi perbaikan proses dapat dilakukan dengan mengidentifikasi pokok pokok permasalahan yang dihadapi oleh bagian ABC berdasarkan Laporan Kinerja 2018.

4. ANALISIS DAN REKOMENDASI

4.1. Keterkaitan Visi, Misi, Tujuan, dan Sasaran Strategis Bagian ABC Instansi XYZ dengan Peraturan Presiden Nomor 95 Tahun 2018 Tentang SPBE dan *Framework* COBIT 5

Pada tahun 2018, Presiden Republik Indonesia mengeluarkan Peraturan Presiden Nomor 95 Tahun 2018 Tentang SPBE. Peraturan ini mendefinisikan bahwa SPBE adalah penyelenggaraan pemerintahan yang memanfaatkan teknologi informasi dan komunikasi untuk memberikan layanan kepada pengguna SPBE. Pengguna SPBE adalah Instansi Pusat, Pemerintah Daerah, pegawai Aparatur Sipil Negara, perorangan, masyarakat, pelaku usaha, dan pihak lain yang memanfaatkan layanan SPBE.

Dari definisi di atas, maka dapat ditarik kesimpulan bahwa bagian ABC merupakan salah satu instansi pemerintah yang menyelenggarakan SPBE. Hal tersebut telah tertuang dalam subbab 4.1.4. yang menjelaskan bahwa Visi, Misi, Tujuan, dan Sasaran Strategis ABC 2015-2019 berfokus pada penyediaan sistem *e-Government* yang aman dan handal.

Berdasarkan Peraturan Presiden Nomor 95 Tahun 2018 Tentang SPBE pada Bab VII Pasal 70 Ayat 1 menyatakan bahwa penyelenggaraan SPBE di suatu instansi pemerintahan harus dipantau dan dievaluasi untuk mengukur dan meningkatkan kualitas SPBE. Oleh karena itu, semua instansi pemerintahan yang memanfaatkan penggunaan TI dalam proses bisnisnya termasuk ABC memiliki urgensi untuk melakukan evaluasi pemanfaatan TI dalam penyelenggaraan SPBE yang sudah diterapkan.

Berdasarkan Peraturan Presiden Nomor 95 Tahun 2018 Tentang SPBE pada Bab VII Pasal 71 Ayat 1, pemantauan dan evaluasi penyelenggaran SPBE di ABC seharusnya didasarkan pada pedoman evaluasi SPBE. Akan tetapi, berdasarkan hasil wawancara, Kepala Bidang Penyiapan Perumusan Kebijakan Penerapan Sistem Pemerintahan Berbasis Elektronik (SPBE) Kementerian Pendayagunaan Aparatur Negara dan Reformasi Birokrasi menyetujui jika pedoman evaluasi SPBE sudah tidak relevan dengan Peraturan Presiden Nomor 95 Tahun 2018 tentang SPBE karena perbedaan pendefinisian ruang lingkup SPBE.

Kepala Bidang Penyiapan Perumusan Kebijakan Penerapan Sistem Pemerintahan Berbasis Elektronik (SPBE) berpendapat bahwa apabila pedoman evaluasi SPBE digunakan untuk mengevaluasi pemanfaatan TI dalam penyelenggaran SPBE di ABC maka tidak semua ruang lingkup SPBE pada Peraturan Presiden Nomor 95 Tahun 2018 akan dinilai. Kepala Bidang Penyiapan Perumusan Kebijakan Penerapan Sistem Pemerintahan Berbasis Elektronik (SPBE) juga memberikan penjelasan bahwa pedoman evaluasi SBPE mengadopsi salah satu *framework* yang sudah diakui secara internasional yaitu *framework* CMMI (*Capability Maturity Model Integration* atau Integrasi Model Kematangan Kemampuan) yang menjadi dasar model tingkat kematangan COBIT 5. Adapun yang diadopsi ialah acuan terkait penentuan level kapabilitas proses SBPE dan beberapa lingkup proses pada COBIT 5 yang disesuaikan dengan ruang lingkup instansi pemerintahan agar menjadi relevan ketika diimplementasikan.

Oleh karena itu, Kepala Bidang Penyiapan Perumusan Kebijakan Penerapan Sistem Pemerintahan Berbasis Elektronik (SPBE) berpendapat bahwa evaluasi pemanfaatan TI dalam penyelenggaraan SPBE dapat dilakukan menggunakan *framework* COBIT 5 dengan syarat tertentu. Syarat yang harus diperhatikan ialah pemetaan antara ruang lingkup SPBE ke dalam *framework* COBIT 5 harus memiliki persamaan ruang lingkup, definisi, maupun tujuan. Pemetaan ini harus dilakukan dengan hati-hati karena bisa jadi satu ruang lingkup SPBE tidak hanya menggambarkan satu ruang lingkup *framework* COBIT 5 saja.

4.2. Pemetaan Ruang Lingkup SPBE ke *Framework* COBIT 5

Ruang lingkup SPBE berdasarkan Peraturan Presiden Nomor 95 Tahun 2018 yang akan menjadi titik fokus pada penulisan Tugas Akhir ini ialah bagian Tata Kelola SPBE dan Manajemen SPBE saja. Ruang lingkup SPBE yang terpilih selanjutnya dipetakan ke dalam *framework* COBIT 5 sebagai sebuah IT *Process* yang akan dinilai menggunakan metode PAM untuk mengetahui pencapaian kapabilitas tata kelola TI bagian ABC di XYZ. Pemetaan ruang lingkup SPBE dilakukan berdasarkan adanya persamaan definisi maupun tujuan antara ruang lingkup SPBE dan *IT Process* COBIT 5. Pemetaan ini merupakan hasil diskusi dengan Kepala Bidang Penyiapan Perumusan Kebijakan Penerapan Sistem Pemerintahan Berbasis Elektronik (SPBE) dan *expert judgment* dari Government Relation and Advocacy Director ISACA Indonesia.

Adapun pemetaan antara ruang lingkup SPBE ke *IT Process framework* COBIT 5 dirangkum dalam Tabel 2.

Tabel 2. Contoh Hasil Pemetaan Ruang Lingkup SPBE ke COBIT 5 [Olahan Sendiri]

No	Ruang Lingkup SPBE	Ruang Lingkup COBIT 5
Unsur Tata Kelola		
1	Rencana Induk SPBE	EDM01
2	Arsitektur SPBE	APO03
3	Peta Rencana SPBE	APO02
4	Rencana dan Anggaran SPBE	APO06
5	Proses Bisnis	APO01
6	Data dan Informasi	APO03
7	Infrastruktur SPBE	APO03
8	Aplikasi SPBE	APO03
9	Keamanan SPBE	APO13
10	Layanan SPBE	APO03
Unsur Manajemen		
11	Manajemen Risiko	APO12
		EDM03
12	Manajemen Kemanan Informasi	APO13
13	Manajemen Data	APO03
14	Manajemen Aset Teknologi Informasi dan Komunikasi	BAI09
15	Manajemen Sumber Daya Manusia	APO07
16	Manajemen Pengetahuan	BAI08
17	Manajemen Perubahan	BAI06
18	Manajemen Layanan SPBE	DSS01
		DSS02
		DSS03
		DSS04

Berdasarkan hasil pemetaan di atas, dapat ditarik sebuah kesimpulan bahwa 18 ruang lingkup SPBE pada aspek tata kelola dan manajemen terpetakan ke 16 dari 37 IT process COBIT 5.

4.3. Penilaian Kapabilitas Tata Kelola TI ABC

IT Process COBIT 5 yang telah terpilih akan menjadi ruang lingkup tata kelola TI di ABC yang dinilai tingkat kapabilitasnya menggunakan metode PAM. Penilaian kapabilitas ini terbagi menjadi dua yaitu dimensi proses dan dimensi kapabilitas. Penilaian pada setiap proses akan berpedoman pada *rating scale* yang telah disepakati dalam memberikan pernyataan pencapaian proses atribut implementasi tata kelola TI di ABC.

Penialian proses pada level 2 hingga level 5 akan bersifat umum untuk semua *IT process* yang terpilih berdasarkan indikator pertanyaan sudah tertera pada *template* metode PAM menggunakan klasifikasi kategori yang sudah disepakati. Akan tetapi, penilaian proses pada level satu bersifat khusus yaitu dengan menilai pencapaian *outcome* setiap *IT process* yang terpilih berdasarkan *base practice* dan *work product*. Pencapaian *base practice* akan dinilai menggunakan kalsifikasi kategori yang sudah disepakati sedangkan pencapaian *work product* akan dinilai melalui hasil *checklist* bukti *work product* pada setiap *IT process*.

Pencapaian suatu level kapabilitas akan terpenuhi jika proses mencapai rating *scale* L atau F. Namun, untuk melanjutkan ke level kapabilitas berikutnya, proses harus mencapai rating *scale* F pada level kapabilitas sebelumnya.

Adapun contoh mekanisme penilaian tingkat kapabilitas pada salah satu *IT Process* diberikan pada Tabel 3 hingga Tabel 6 dan hasil penilaian tingkat kapabilitas tata kelola TI di ABC diberikan pada Tabel 7.

Tabel 3. Contoh Capaian Kapabilitas proses EDM01 [Olahan Sendiri]

Process Name	Level 0	Level 1	Level 2		Level 3		Level 4		Level 5	
EDM01		PA 1.1	PA 2.1	PA 2.2	PA 3.1	PA 3.2	PA 4.1	PA 4.2	PA 5.1	PA 5.2
Rating Percent		93%	80%	80%						
Rating by Criteria		F	L	L						
Capability Level Achieved				2						

Tabel 4. Contoh Capaian Penilaian Kapabilitas pada PA 1.1 proses EDM01 [Olahan Sendiri]

EDM01	Penilaian Capaian *Outcomes*	Kriteria	Y/N	Pencapaian			
				N	P	L	F
Level 1 *Performed*	PA 1.1 - *Process Performance* Proses mencapai tujuan proses	**EDM01-O1** Model pengambilan keputusan strategis untuk TI dilakukan secara efektif dan selaras dengan kondisi lingkungan internal dan eksternal serta persyaratan pemangku kepentingan organisasi	Y				100 %
		EDM01-O2 Penerapan sistem tata kelola TI di organisasi					93%
		EDM01-O3 Perolehan jaminan bahwa sistem tata kelola TI beroperasi secara efektif				80%	

Tabel 5. Contoh Capaian Penilaian Kapabilitas pada PA 2.1 proses EDM01 [Olahan Sendiri]

EDM01	Penilaian Capaian *Outcomes*	Kriteria	Y/N	Pencapaian			
				N	P	L	F
Level 2 *Managed*	PA 2.1 – *Performance Management* Kinerja telah dikelola	Sasaran kinerja proses telah diidentifikasi	Y				100 %
		Kinerja proses telah direncanakan dan dipantau				70%	
		Kinerja proses disesuaikan agar selaras dengan rencana				70%	
		Tanggung jawab dan otoritas untuk melaksanakan proses telah diidentifikasi dan dikomunikasikan					100 %
		Sumber daya dan informasi yang dibutuhkan untuk melaksanakan kinerja proses telah diidentifikasi, tersedia, dialokasikan dan digunakan				70%	
		Tatap muka pihak yang terlibat telah dikelola untuk memastikan komunikasi yang efektif dan tanggung jawab yang jelas				70%	

Tabel 6. Contoh Capaian Penilaian Kapabilitas pada PA 2.2 proses EDM01 [Olahan Sendiri]

EDM01	Penilaian Capaian *Outcomes*	Kriteria	Y/N	Pencapaian			
				N	P	L	F
Level 2 *Managed*	PA 2.1 – *Work Product Management* *Work product* telah dikelola	Persyaratan untuk *work product* dari kinerja proses telah diidentifikasi	Y			80%	
		Persyaratan dokumentasi dan kontrol dari *work product* telah diidentifikasi				70%	
		Work product telah diidentifikasi, terdokumentasi, dan terkontrol					100 %
		Work product di *review* agar selaras dengan perencanaan dan persyaratan				70%	

Tabel 7. Hasil Penilaian Tata Kelola ABC [Olahan Sendiri]

No	IT Process	Level	Hasil
1	Ensure governance setting and maintenance	2	84%
2	Ensure risk operations	1	68%
3	Manage the IT management framework	1	83%
4	Manage strategy	1	85%
5	Manage enterprise architecture	1	52%
6	Manage budget and cost	2	83%
7	Manage human resource	1	69%
8	Manage risk	1	67%
9	Manage security	1	82%
10	Manage change	1	65%
11	Manage knowledge	1	56%
12	Manage asset	1	81%
13	Manage operation	2	67%
14	Manage service requests and incidents	1	81%
15	Manage problem	1	70%
16	Manage continuity	1	59%

4.4. Analisis Kesenjangan

Berdasarkan hasil wawancara dengan Kepala ABC selaku penanggung jawab tertinggi pengelolaan TI di XYZ diperoleh informasi bahwa penilaian tingkat kapabilitas tata kelola TI menggunakan metode *Process Assesment Model* (PAM) COBIT 5 di ABC diharapkan agar berada pada tingkat dua (*Managed Process*) yaitu proses TI telah direncanakan, dikelola, dipantau, dan disesuaikan, serta *work product* dari proses TI disusun dengan baik, dikontrol, dan dipelihara seoptimal mungkin.

Analisis kesenjangan dilakukan dengan cara membuat perbandingan antara penetapan target yang sudah ditetapkan oleh Kepala ABC dan hasil penilaian tingkat kapabilitas dari setiap proses yang telah didefinisikan menggunakan metode PAM yang digunakan sebagai informasi untuk mengetahui upaya apa yang dibutuhkan untuk mencapai target yang ditetapkan melalui sebuah rekomendasi perbaikan. Hasil nilai kesenjangan tingkat kapabilitas proses tata kelola TI di ABC berdasarkan ruang lingkup SPBE menggunakan *framework* COBIT 5 dirangkum dalam Tabel 8.

Tabel 8. Hasil Analisis Kesenjangan [Olahan Sendiri]

Proses COBIT 5			Tingkat Kapabilitas		Nilai Kesenjangan
			Hasil Penilaian	Target	
Evaluate, Direct, and Monitor	EDM01	*Ensure Governance Framework Setting and Maintenance*	2	2	0
	EDM03	*Ensure Risk Optimation*	1	2	1
Align, Plan, and Organise	APO01	*Manage The IT Management Framework*	1	2	1
	APO02	*Manage Strategy*	1	2	1
	APO03	*Manage Enterprise Architecture*	1	2	1
	APO06	*Manage Budget and Cost*	2	2	0
	APO07	*Manage Human Resource*	1	2	1
	APO12	*Manage Risk*	1	2	1
	APO13	*Manage Security*	1	2	1
Build, Acquire, and Implement	BAI06	Manage Change	1	2	1
	BAI08	*Manage Knowledge*	1	2	1
	BAI09	*Manage Asset*	1	2	1
Deliver, Service, and Suppot	DSS01	*Manage Operation*	1	2	1
	DSS02	*Manage Service Requests and Incidents*	1	2	1
	DSS03	*Manage Problem*	1	2	1
	DSS04	*Manage Continuity*	1	2	1

Dari hasil kesenjangan ini dapat disimpulkan bahwa terdapat 2 IT Process yang sudah mencapai target kapabilitas yaitu EDM01 dan APO06.

4.5. Rekomendasi

Rekomendasi peningkatan implementasi tata kelola TIK ABC akan dibuat berdasarkan proses atribut metode *Process Assessment Model* (PAM) COBIT 5. Mekanisme pemberian rekomendasi akan dilakukan pada setiap tingkat hingga proses tersebut mencapai kategori *fully achieved* (F), kemudian dapat mengaplikasikan rekomendasi pada tingkat berikutnya.

Salah satu faktor kunci keberhasilan implementasi COBIT 5 dalam organisasi yaitu bagaimana cara organisasi memprioritaskan perbaikan proses dengan mempertimbangkan aspek yang dinilai paling bermanfaat dan paling mudah diwujudkan. Pemrioritasan perbaikan proses dapat dilakukan dengan mengidentifikasi pokok-pokok permasalahan yang dihadapi oleh organisasi dan peristiwa-peristiwa penting yang menjadi pemicu diperlukannya perbaikan proses [7].

Pokok permasalahan spesifik pengelolaan TI di ABC diidentifikasi dari dokumen Laporan Kinerja ABC Tahun 2018 yang divalidasi oleh Kepala ABC untuk mengetahui apa saja permasalahan pengelolaan TI paling strategis yang dihadapi ABC. Berdasarkan Laporan Kinerja Tahun 2018, ABC mengalami kendala pengelolaan TI yang difokuskan dalam empat permasalahan utama. Dari empat permasalahan utama ini, akan diidentifikasi apa saja kondisi yang menyebabkan dan memicu permasalahn tersebut. Hasil identifikasi permasalahan pengelolaan TI di ABC kemudian akan dihubungkan ke *IT process* COBIT 5 yang relevan dalam menentukan pemrioritasan rekomendasi perbaikan proses.

Adapun rangkuman hasil identifikasi pokok permasalahan pengelolaan TI di ABC ke IT process COBIT 5 ialah sebagai berikut:

Tabel 9. Rangkuman Hasil Pemetaan Identifikasi Masalah [Olahan Sendiri]

No	IT Process	Jumlah Masalah
1	EDM01	5
2	EDM03	1
3	APO01	2
4	APO02	9
5	APO03	5
6	APO06	4
7	APO07	3
8	APO12	4
9	APO13	2
10	BAI06	3
11	BAI08	-
12	BAI09	3
13	DSS01	2
14	DSS02	4
15	DSS03	2
16	DSS04	5

Berdasarkan tabel di atas, penentuan pemrioritasan rekomendasi perbaikan *IT process* pengelolaan TI di ABC yang disarankan untuk dilakukan terlebih dahulu yaitu pada proses APO02, APO03, APO07, DSS04, APO12 dan DSS02 dengan mempertimbangkan bahwa implementasi proses EDM01 dan APO06 di ABC telah mencapai level 2. Contoh rekomendasi yang diberikan disajikan dalam Tabel 10.

Tabel 10. Contoh Rekomendasi pada Proses APO02 [Olahan Sendiri]

No	*IT Process*	Rekomendasi
1	APO02	Melakukan evaluasi dan analisis lebih lanjut untuk menentukan kebutuhan perubahan proses bisnis di PusTIK KP
		Melakukan pemantauan dan evaluasi terhadap kebijakan, prosedur, aktivitas, program, maupun praktik terbaik terkait TI agar dapat menjamin keselarasan dan kesesuaian dengan sasaran strategi TI maupun sasaran strategi organisasi
		Menyusun sasaran strategi TI PusTIK KP terbaru dengan memperhatikan faktor internal maupun eksternal organisasi melalui analisis SWOT agar dapat melakukan perencanaan dan penetapan target TI yang tepat dengan memperhatikan faktor anggaran yang tersedia.
		Membuat dokumentasi resmi semua persyaratan dan kontrol *work product* untuk dijadikan pedoman dalam menentukan pencapaian *outcome* maupun *output* strategi TI PusTIK KP.
		Memantau secara berkala pencapaian *outcome* maupun *output work product* strategi TI PusTIK KP.

5. SIMPULAN DAN SARAN

Penilaian tingkat kapabilitas tata kelola TI di ABC yang dilakukan pada IT *process* yang terpilih menggunakan metode PAM menghasilkan 14 IT *process* berada pada level 1 (*performed*) dengan capaian indikator PA 1.1 *largely achieved* dan 2 IT *process* berada pada level 2 (*managed*) dengan capaian indikator PA 2.1 dan PA 2.2 *largely achieved*. Hasil analisis kesenjangan menunjukkan bahwa terdapat 2 IT *process* yang telah memenuhi target level yang ditetapkan Kepala ABC yaitu EDM01 dan APO06. Rekomendasi disusun berdasarkan proses atribut metode PAM COBIT 5 agar dapat mencapai kategori *fully achieved* pada level 1 maupun level 2 dan masalah prioritasi rekomendasi yang diberikan pada proses APO02, APO03, APO07, DSS04, APO12 dan DSS02.

Adapun saran yang perlu diperhatikan pada penelitian kali ini adalah sebagai berikut:

a. ABC diharapkan mampu menyempurnakan pelaksanaan semua tata kelola TI yang sudah ada dan melengkapi yang belum ada.
b. ABC diharapkan mampu memelihara kepabilitas yang dimiliki saat ini dalam menetapkan standar dan prosedur untuk setiap proses, mendefinisikan tanggung jawab dan kewenangan dalam pengelolaan proses TI, melakukan sosialisasi mengenai tata kelola TI, membudayakan pelaporan dan evaluasi berkala.
c. Penelitian ini diharapkan mampu menjadi bahan masukan untuk penelitian selanjutnya terkait perlunya revisi pedoman evaluasi SPBE.

Referensi

[1] Republik Indonesia 2018. Peraturan Presiden Nomor 95 Tahun 2018 tentang Sistem Pemerintahan Berbasis Elektronik. Indonesia: Jakarta.

[2] M Weber, Ron. 1999. Information System Control Audit. New Jersey: Prentice Hall

[3] Hunton, J.E, et al. 2004. Core Concepts of Information Technology Auditing. Amerika Serikat: John W & Sons, Inc.

[4] Putra, Hervandi. 2014. Penerapan dan Penilaian Tata Kelola Teknologi Informasi berdasarkan COBIT 5 Framework. Depok: Universitas Indonesia

[5] Bouty, A.A., et al. 2018. New Model of Information Technology Governance in the Government of Gorontalo City using Framework COBIT 4.1. IOP Conference Series: Materials Science and Engineering.

[6] XYZ. 2016. Peraturan Menteri XYZ Republik Indonesia Nomor 2 Tahun 2016 Tentang Organisasi dan Tata Kerja XYZ. Indonesia: Jakarta.

[7] ISACA, 2012. A Business Framewrok for The Governance and Management of Enterprise IT. USA: ISACA.

[8] Radovanovic, D, et al. 2010. Information Technology Governance – COBIT Model.

[9] Oltsik, Jon. 2003. Information Security Management: Analysis andd Recommendations. Inggris: Hype-free Consulting

[10] Gondodiyoto, S. 2007. Audit Sistem Informasi dengan Pendekatan COBIT. Jakarta: Mitra Wacana Media.

[11] Information Technology Governance Institute (ITGI). 2007. COBIT 4.1 Executive Summary. USA: ITGI.

[12] Information Technology Governance Institute (ITGI). 2003. Board Briefing on IT Governance. USA: ITGI.

[13] ISACA. 2013. Process Assesment Model (PAM): Using COBIT 5. USA: ISACA.

[14] Yukov, B. 2006. Info-Gap Decision Theory (Second Edition).

[15] Regina, Oliva. 2017. Perancangan Tata Kelola Keamanan Informasi di Pusdatin Kemenhan Menggunakan Framework COBIT 5 for Information Security. Bogor: Sekolah Tinggi Sandi Negara.

[16] Irawan, Prasetya. 2003. Logika dan Prosedur Penelitian: Pengantar Teori dan Panduan Praktis Penelitian Sosial bagi Mahasiswa dan Peneliti Pemula. Jakarta: STIA-LAN Press.

Kriptanalisis Integral pada SKINNY-64-64 Putaran Tereduksi

Nurul Aisyah[1)] dan Andriani Adi Lestari[2)]
(1) Politeknik Siber dan Sandi Negara / nurul.aisyah@student.polteksn.ac.id
(2) Politeknik Siber dan Sandi Negara / aaltari@gmail.com

Abstrak

SKINNY-64-64 merupakan algoritme lightweight tweakable block cipher *yang didesain oleh Beierle et al. pada tahun 2016. Salah satu teknik kriptanalisis yang diterapkan oleh Beierle et al. adalah kriptanalisis integral. Beierle et al. melakukan kriptanalisis integral empat belas putaran menggunakan* integral distinguisher *5 putaran dengan empat putaran proses* backward direction *dan dapat mengekstraksi enam sel masukan* tweakey. *Pada penelitian ini, dilakukan pencarian* integral distinguisher *menggunakan himpunan teks terang dengan mengaktifkan satu hingga empat sel. Berdasarkan hasil penelitian,* integral distinguisher *terbaik yang ditemukan adalah* integral distinguisher *7 dan 7,5 putaran sebanyak 744.* Integral distinguisher *7 dan 7,5 putaran digunakan untuk mendesain skenario perluasan serangan hingga putaran ke-12. Skenario kriptanalisis integral pada SKINNY-64-64 12 putaran terbaik pada penelitian ini dapat mengekstraksi delapan sel masukan* tweakey *dengan membutuhkan kompleksitas data, memori, dan waktu yaitu* $2^{12}, 2^{12}$, *dan* 2^{44} *secara berturut-turut. Sel* tweakey *lain diekstraksi dengan menggunakan serangan* brute force *sehingga kompleksitas waktu total adalah* $2^{44} + 2^{32}$. *Serangan tersebut dilakukan menggunakan* integral distinguisher *terbaik dengan tiga sel aktif pada posisi (4,11,14),(5,8,15),(6,9,12) dan (7,10,13).*

Kata Kunci: *integral distinguisher* (1), *kriptanalisis integral* (2), *SKINNY-64-64* (3), *tweakable block cipher* (4).

1. PENDAHULUAN

Tweakable block cipher merupakan varian baru dari kriptografi primitif yang bertujuan untuk membuat algoritme *block cipher* dengan performa yang baik dan tidak membutuhkan banyak biaya. Secara umum, *tweakable block cipher* memiliki prinsip yang sama dengan *block cipher*, yaitu algoritme enkripsi dan dekripsi yang memperlakukan blok masukan secara keseluruhan untuk menghasilkan blok keluaran dengan panjang sama [1]. Namun, pada *tweakable block cipher*, masukan yang dibutuhkan bukan hanya pesan dan kunci, tetapi juga masukan ketiga yang dinamakan "*tweak*" [2]. *Tweak* tersebut digunakan untuk memilih suatu permutasi. *Tweak* memiliki fungsi serupa dengan *Initialization Vector* pada mode operasi *Cipher Block Chaining* (CBC) atau *nonce* pada mode operasi *Offset Codebook* (OCB). Salah satu algoritme *tweakable block cipher* adalah algoritme SKINNY [3].

Algoritme SKINNY merupakan *lightweight tweakable block cipher* dengan struktur *Subtitution Permutation Network* (SPN). SKINNY diklaim memiliki performa perangkat lunak ataupun keras yang lebih baik dengan keamanan yang lebih kuat dibandingkan dengan algoritme SIMON [4] karena SKINNY menggunakan hanya sedikit gerbang logika sehingga sangat efisien untuk diimplementasikan. Pada SKINNY, *tweak* dan kunci diperlakukan sama, dan disebut sebagai *tweakey*. *Framework tweakey* tersebut diperkenalkan oleh Jean *et al.*[5].

SKINNY merupakan algoritme *tweakable block cipher* baru sehingga belum banyak serangan yang diterapkan pada algoritme tersebut. Pada analisis keamanan SKINNY, Beierle *et al.* dalam [3] menjelaskan beberapa serangan yang dilakukan pada algoritme tersebut. Salah satu serangan yang diterapkan pada algoritme SKINNY adalah kriptanalisis integral yaitu teknik kriptanalisis yang memerhatikan propagasi jumlah dari beberapa nilai setelah melewati proses enkripsi pada jalur tertentu [6]. Menurut Z'aba *et al.* dalam [7], ide dasar dari serangan ini adalah untuk menganalisis sifat spesifik pada himpunan teks terang setelah melalui proses enkripsi dan menggunakan keberadaan sifat tersebut untuk memverifikasi kunci yang ditebak. Kriptanalisis integral terdiri dari dua tahapan, yaitu pencarian *integral distinguisher* dan proses ekstraksi kunci (*key recovery*) [8].

Beierle *et al.* melakukan pencarian *integral distinguisher* pada dua varian algoritme SKINNY, salah satunya yaitu pada varian SKINNY-64-64. Pencarian *integral distinguisher* oleh Beierle *et al.* dilakukan dengan mengaktifkan satu sel berukuran empat bit, kemudian dilakukan enkripsi hingga semua sel bersifat *Unknown* (*U*). Beierle *et al.* menggunakan *integral distinguisher* satu kemungkinan posisi sel aktif yaitu pada sel di baris kedua kolom pertama.

Beierle *et al.* menerapkan suatu metode yang menghasilkan sifat integral berorde tinggi yaitu dengan cara melakukan propagasi satu sel aktif pada putaran kelima. Selanjutnya, Beierle *et al.* melakukan proses *backward direction* algoritme SKINNY-64-64 sebanyak empat putaran sehingga menghasilkan dua belas sel aktif pada putaran pertama dan *integral distinguisher* 10 putaran pada algoritme SKINNY-64-64. Beierle *et al.* menyatakan dari *integral distinguisher* 10 putaran algoritme SKINNY-64-64 yang dihasilkan dapat digunakan untuk mengekstraksi *subtweakey* pada putaran kedua belas, ketiga belas, dan keempat belas. Kompleksitas data yang dibutuhkan oleh Beierle *et al*. untuk melakukan

serangan ini adalah 2^{48} teks terang terpilih dan membutuhkan akses memori untuk 2^{48} teks sandi.

Berdasarkan hal tersebut, pada penelitian ini dilakukan pencarian *integral distinguisher* terbaik pada algoritme SKINNY-64-64 dengan mengaktifkan maksimum empat sel berukuran empat bit pada semua kemungkinan posisinya. Kriptanalisis integral pada algoritme SKINNY-64-64 dilakukan dengan menggunakan *integral distinguisher* terbaik yang ditemukan. Pencarian *integral distinguisher* dilakukan tanpa proses *backward direction. Integral distinguisher* terbaik yang ditemukan selanjutnya digunakan untuk membuat desain skenario serangan dengan kompleksitas paling efektif.

Pembahasan hasil penelitian ini disusun menjadi 5 (lima) bagian. Pada bagian pertama dibahas latar belakang penelitian. Teori-teori yang mendasari penelitian dijelaskan pada bagian dua. Bagian tiga dibahas metodologi penelitian. Bagian inti dari penelitian ini yaitu penerapan kriptanalisis integral pada SKINNY-64-64 putaran tereduksi. Kesimpulan dan saran untuk penelitian selanjutnya dijelaskan pada bagian lima dan bagian enam berisi daftar pustaka yang menjadi rujukan dari penelitian ini.

2. LANDASAN TEORI

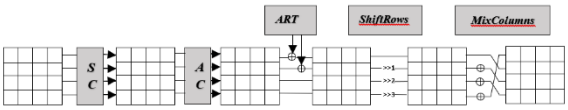
2.1. Algoritme SKINNY-64-64

Penjelasan mengenai algoritme SKINNY-64-64 seluruhnya merujuk pada [3]. SKINNY adalah algoritme *lightweight tweakable block cipher* yang diajukan oleh Beierle *et al.* pada CRYPTO 2016. Algoritme ini memiliki struktur *Substitution-Permutation Network* (SPN) dengan dua jenis varian ukuran blok n yaitu 64 dan 128 bit. SKINNY menggunakan kerangka *tweakey* seperti konstruksi yang diajukan oleh Jean *et al*. pada tahun 2014 [5]. Algoritme *tweakable block cipher* dengan kerangka *tweakey* yaitu algoritme yang mengganti kunci masukan atau pasangan kunci/*tweak* dengan menggunakan *tweakey* (TK). *Tweakey* pada algoritme SKINNY terdiri dari tiga varian ukuran, yaitu $t = n$, $t = 2n$, dan $t = 3n$. Varian algoritme SKINNY dapat dituliskan dengan SKINNY-n-t. Pada penelitian ini, varian algoritme SKINNY yang digunakan memiliki ukuran blok dan *tweakey* 64 bit atau dituliskan dengan SKINNY-64-64.

Pada SKINNY-64-64, teks terang direpresentasikan sebagai $IS = is_0 \parallel is_1 \parallel \cdots \parallel is_{14} \parallel is_{15}$, kemudian dimasukkan ke dalam *internal state IS* seperti pada matriks berukuran 4×4 berikut:

$$IS = \begin{bmatrix} is_0 & is_1 & is_2 & is_3 \\ is_4 & is_5 & is_6 & is_7 \\ is_8 & is_9 & is_{10} & is_{11} \\ is_{12} & is_{13} & is_{14} & is_{15} \end{bmatrix}.$$

Proses enkripsi SKINNY-64-64 menerapkan fungsi putaran yang diiterasi sebanyak 32 putaran. Proses dekripsi dilakukan dengan menggunakan invers dari fungsi putaran. Fungsi putaran pada SKINNY-64-64 terdiri dari lima operasi dengan urutan *SubCells*, *AddConstants*, *AddRoundTweakey*, *ShiftRows*, dan *MixColumns*, seperti yang ditunjukkan pada Gambar 1.



Gambar 1. Fungsi putaran pada SKINNY-64-64 [3]

Fungsi *SubCells* (*SC*) memetakan *IS* secara bijektif oleh *s-box* seperti yang ditunjukkan pada Tabel 1. Pada fungsi *AddConstants* (*AC*), *IS* di-XOR-kan dengan fungsi putaran algoritme SKINNY-64-64 (Lihat Tabel 2). *Subtweakey* di-XOR-kan dengan *IS* pada proses *AddRoundTweakey* (*ART*). Proses enkripsi dilanjutkan dengan fungsi *ShiftRows* yang merotasi *IS* dengan ketentuan seperti ditunjukkan pada Gambar 2. Fungsi putaran terakhir yaitu *MixColumns* yang mengalikan setiap kolom pada *IS* dengan matriks biner *M* sebagai berikut:

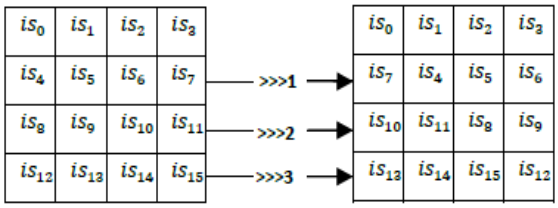
$$M = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}.$$

Tabel 1 *S-box* dari SKINNY-64-64 [3]

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
s	c	6	9	0	1	a	2	b	3	8	5	d	4	e	7	f

Tabel 2 Konstanta putaran SKINNY-64-64 [3]

Round	Konstanta (heksadesimal)
1-16	01,03,07,0f,3e,3d,3b,37,2f,1e,3c,39,33,27,0e
17-32	1d,3a,35,2b,16,2b,18,30,21,02,05,0b,17,2e,1c



Gambar 2 Fungsi *ShiftRows* pada SKINNY-64-64

Subtweakey pada SKINNY-64-64 dihasilkan oleh algoritme penjadwalan *tweakey* dengan masukan berupa *tweakey* berukuran 64 bit. Masukan *tweakey* dinotasikan sebagai $TK = tk_0 \parallel tk_1 \parallel \cdots \parallel tk_{15}$ yang direpresentasikan dalam bentuk matriks 4×4

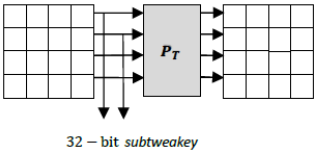
$$TK = \begin{bmatrix} tk_0 & tk_1 & tk_2 & tk_3 \\ tk_4 & tk_5 & tk_6 & tk_7 \\ tk_8 & tk_9 & tk_{10} & tk_{11} \\ tk_{12} & tk_{13} & tk_{14} & tk_{15} \end{bmatrix}.$$

Algoritme ini menggunakan *array* $TK1$ untuk membentuk *subtweakey* karena memiliki nilai *z* sama

dengan satu $\left(z = \frac{t}{n} = \frac{64}{64} = 1\right)$. Notasi $TK1^r = tk1_0^r, \ldots, tk1_{15}^r$ untuk $1 \leq r \leq 32$ menyatakan *array* $TK1$ pada putaran r. Nilai awal dari *array* $TK1^1 = TK$. Proses penjadwalan *tweakey* untuk $1 \leq r \leq 32$ dilakukan sebagai berikut:

a. *subtweakey* pada putaran ke-r adalah $tk1_0^r, \ldots, tk1_7^r$;
b. $TK1^r$ diperbarui menggunakan fungsi permutasi $P_T = [9,15,8,13,10,14,12,11,0,1,2,3,4,5,6,7]$ untuk putaran selanjutnya atau dengan kata lain $TK1^{r+1} = P_T(TK1^r)$.

Gambar 3 menunjukkan proses penjadwalan *tweakey* pada SKINNY-64-64.



Gambar 3 Penjadwalan *tweakey* SKINNY-64-64 [3]

2.2. Kriptanalisis Integral

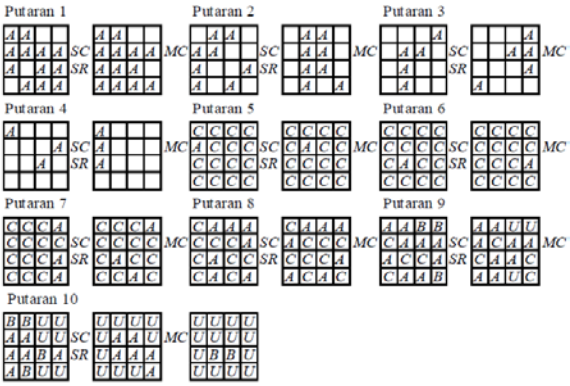
Kriptanalisis integral merupakan teknik kriptanalisis yang memerhatikan propagasi jumlah dari beberapa nilai setelah melewati proses enkripsi pada jalur tertentu [6]. Kriptanalisis integral termasuk dalam *chosen plaintext attack* sehingga untuk melakukan serangan ini, penyerang membangkitkan himpunan teks terang yang memiliki nilai konstan pada sejumlah sel, sedangkan sel lainnya terdiri dari semua nilai [6]. Serangan ini terdiri dari dua tahapan, yaitu melakukan pencarian *integral distinguisher* dan melakukan ekstraksi kunci menggunakan *integral distinguisher* yang telah diperoleh [8].

Pada kriptanalisis integral terdapat dua metode, yaitu metode orde rendah (*low order*) dan orde tinggi (*higher order*) [6]. Todo dalam [9] mendefinisikan sifat integral (*integral property*) yang terdiri dari empat sifat, yaitu sebagai berikut:

All (*A*) : semua kemungkinan nilai pada sel muncul dengan jumlah yang sama.
Constant (*C*) : sel memiliki nilai sama pada seluruh anggota himpunan teks terang.
Balanced (*B*) : hasil penjumlahan dari semua nilai sel dalam himpunan teks terang sama dengan nol.
Unknown (*U*) : hasil penjumlahan dari semua nilai sel dalam himpunan teks terang tidak sama dengan nol.

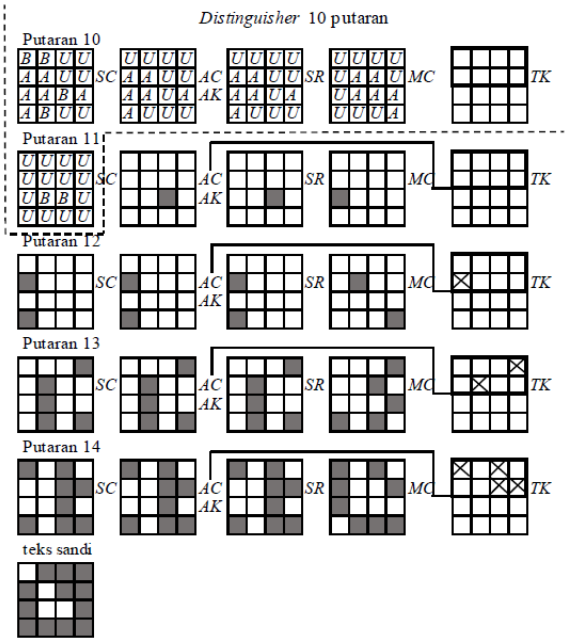
2.3. Kriptanalisis Integral pada SKINNY-64-64 oleh Beierle *et al.*

Pada [3], Beierle *et al*. melakukan pencarian *integral distinguisher* dengan mengaktifkan satu sel, kemudian mengenkripsi sampai dengan semua sel bersifat *U*. *Integral distinguisher* yang berorder tinggi (*higher order*) diperoleh dengan mempropagasi *backward direction* algoritme SKINNY-64-64 sebanyak empat putaran, dan menghasilkan dua belas sel aktif pada masukan putaran pertama dan *integral distinguisher* pada sepuluh putaran algoritme SKINNY-64-64 seperti ditunjukkan pada Gambar 4.



Gambar 4 *Integral distinguisher* 10 putaran yang ditemukan oleh Beierle *et al.* [3]

Beierle *et al.* menyatakan dari *integral distinguisher* 10 putaran yang dihasilkan dapat dilakukan ekstraksi kunci hingga empat belas putaran algoritme SKINNY-64-64. Kompleksitas data dari serangan kriptanalisis integral yang dilakukan Beierle *et al*. pada algoritme SKINNY-64-64 adalah $2^{12 \cdot 4} = 2^{48}$ teks terang terpilih (*chosen plaintexts*), dan akses memori hingga 2^{48} yang digunakan untuk menyimpan himpunan teks sandi, selain itu pada tahapan pertama membutuhkan memori sebanyak 2^{32} untuk menyimpan nilai *state* setelah proses *parity check*. Perluasan putaran serangan untuk melakukan ekstraksi *tweakey* yang dilakukan oleh Beierle *et al*. ditunjukkan pada Gambar 5.



Gambar 5 Empat belas putaran proses ekstraksi *subtweakey* pada SKINNY-64-64 oleh Beierle *at al.* [3]

3. METODOLOGI PENELITIAN

Metode penelitian yang digunakan pada penelitian ini adalah metode telaah kepustakaan dan metode eksperimen. Metode telaah kepustakaan dilakukan melalui kajian terhadap teori tentang algoritme SKINNY-64-64, kriptanalisis integral, dan kriptanalisis integral pada SKINNY-64-64 oleh Beierle *et al.* menggunakan sumber berupa buku, jurnal dan sumber lainnya yang mendukung penelitian

Metode eksperimen dilakukan melalui pencarian *integral distinguisher* terbaik pada algoritme SKINNY-64-64 dengan mencoba mengaktifkan maksimum empat sel pada semua kemungkinan posisi. Berikut adalah tahapan untuk mencari *integral distinguisher* terbaik pada SKINNY-64-64 dengan mencoba maksimum empat sel aktif pada semua kemungkinan posisi.

a. Menentukan pasangan teks terang IS.
b. Menentukan nilai p yaitu jumlah sel aktif $1 \leq p \leq 4$.
c. Menentukan posisi sel aktif dengan penomoran posisi sel sebagaimana ditunjukkan pada Gambar 6.

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Gambar 6 Penomoran posisi sel algoritme SKINNY-64-64

d. Membangkitkan himpunan teks terang berdasarkan terks terang IS yang dipilih pada poin a.
e. Melakukan enkripsi terhadap himpunan teks terang yang dibangkitkan dengan komponen-komponen fungsi putaran algoritme SKINNY-64-64.
f. Meng-XOR-kan nilai keluaran setiap fungsi putaran yang dilakukan pada setiap sel untuk mengevaluasi sifat integral sel.
g. Apabila hasil XOR sama dengan nol, maka sel masih dalam kondisi seimbang yaitu bersifat *Balanced* (*B*) sehingga proses enkripsi dilanjutkan sampai pada putaran dengan hasil XOR dari semua nilai keluaran pada semua sel tidak sama dengan nol atau disebut *Unknown* (*U*).
h. Tahapan pada poin a. hingga g. diulangi menggunakan semua kemungkinan satu sampai empat posisi sel aktif.
i. Kemudian menentukan *integral distinguisher* terbaik yang ditemukan.

Integral distinguisher terbaik yang diperoleh digunakan dalam mendesain skenario serangan untuk mengetahui kompleksitas dari kriptanalisis integral pada SKINNY-64-64. Di bawah ini dijelaskan tahapan kriptanalisis integral pada SKINNY-64-64 menggunakan *integral distinguisher* terbaik.

a. Melakukan perluasan serangan dengan memperluas putaran *integral distinguisher*.
b. Melakukan enkripsi terhadap himpunan teks terang yang dibangkitkan berdasarkan jumlah sel aktif pada masukan *integral distinguisher*. Enkripsi dilakukan hingga pada putaran yang sudah diperluas.
c. Teks sandi yang dihasilkan dideskripsi secara parsial hingga pada putaran *integral distinguisher*.
d. Hasil dekripsi di-XOR-kan secara per sel. Apabila hasilnya bersifat *B* maka *subtweakey* target sesuai *integral distinguisher* merupakan kandidat *subtweakey* yang benar.
e. Apabila terdapat lebih dari satu kandidat *subtweakey* yang benar, maka langkah b., c., dan d. dapat diulang. *Subtweakey* pada posisi sel lain dapat ditebak dengan serangan *brute force*.

4. PENERAPAN KRIPTANALISIS INTEGRAL PADA SKINNY-64-64 PUTARAN TEREDUKSI

4.1. Pencarian *Integral Distinguisher*

Pada proses pencarian *integral distinguisher*, pasangan teks terang dan *tweakey* yang berbeda membentuk pola *integral distinguisher* yang berbeda sehingga untuk mencari pola *integral distinguisher* yang paling dominan muncul pada penelitian ini digunakan delapan teks terang dan *tweakey* (Lihat Tabel 3). Delapan teks terang dan *tweakey* dikombinasikan untuk membentuk 64 pasangan teks terang dan *tweakey*. Apabila dari 64 pasangan teks terang dan *tweakey* tidak menghasilkan pola *integral distinguisher* yang dominan maka dapat dilakukan penambahan pasangan teks terang dan *tweakey* hingga dapat menemukan pola *integral distinguisher* yang dominan.

Tabel 3 Daftar teks terang dan *tweakey*

No	Teks Terang	*Tweakey*
1	0603 4f95 724 $d19d_{16}$	$f526$ 9826 $fc68$ 1238_{16}
2	$cf16$ $cfe8$ $fd0f$ $98aa_{16}$	$9eb9$ 3640 $d088$ $da63_{16}$
3	$530c$ $61d3$ $5e86$ $63c3_{16}$	$76a3$ $9d1c$ $8bea$ $71e1_{16}$
4	$f20a$ $db0e$ $b08b$ $648a_{16}$	$ed00$ $c85b$ $120d$ 6861_{16}
5	$b86f$ $d759$ $9ea5$ $bae6_{16}$	$cf91$ $6e88$ $273a$ 9192_{16}
6	5167 5016 5533 $7e7b_{16}$	$02c6$ $545c$ $db0d$ $e091_{16}$
7	$32fd$ $a7d7$ $844a$ $70ac_{16}$	$937e$ $cfb0$ $a0c4$ 9802_{16}
8	$4d84$ $801a$ $0bfe$ $df7b_{16}$	$32da$ 8217 $f6c5$ $192b_{16}$

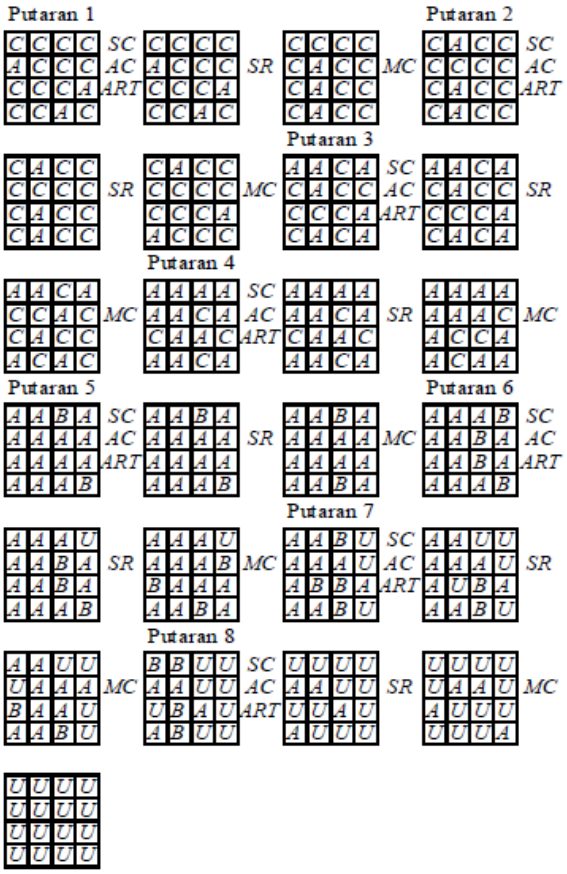
Setelah menentukan pasangan teks terang dan *tweakey*, proses selanjutnya adalah membangkitkan himpunan teks terang dari teks terang yang dipilih. Jumlah anggota himpunan teks terang ditentukan berdasarkan jumlah sel aktif pada masukan putaran pertama.

Jumlah teks terang yang dibangkitkan dalam mencari *integral distinguisher* menggunakan satu sel aktif untuk setiap posisinya membutuhkan 2^4 teks terang. Pada pencarian *integral distinguisher* dengan dua sel aktif membangkitkan teks terang sejumlah 2^8 dengan nilai dari dua sel yang diaktifkan yaitu semua kemungkinan $0 - (2^8 - 1)$. Himpunan teks terang

untuk *integral distinguisher* dengan tiga sel aktif terdiri dari 2^{12} teks terang dan untuk empat sel aktif terdiri dari 2^{16} teks terang untuk setiap kemungkinan posisi.

Himpunan teks terang yang telah dibangkitkan kemudian dienkripsi menggunakan algoritme SKINNY-64-64. Sifat integral pada keluaran setiap komponen dari fungsi putaran dievaluasi. Evaluasi dilakukan dengan meng-XOR-kan setiap hasil keluaran yang dilakukan untuk setiap sel. Proses enkripsi dihentikan ketika keluaran komponen dari fungsi putaran menghasilkan seluruh sel bersifat integral *Unknown* yang dinotasikan dengan *U*. Kemudian, proses pencarian *integral distinguisher* diulang dengan menggunakan pasangan teks terang dan *tweakey* lain. Selanjutnya, pola yang paling dominan muncul saja yang dipilih sebagai *integral distinguisher*.

Pada Gambar 7. ditunjukkan pola *integral distinguisher* yang paling dominan muncul ketika mengaktifkan tiga sel pada posisi (4,11,1).



Gambar 7 *Integral distinguisher* 7,5 putaran dengan satu sel aktif pada posisi (4,11,14)

Berdasarkan *integral distinguisher* pada Gambar 7. seluruh sel bersifat *U* setelah melewati *MixColumns* putaran kedelapan sehingga *integral distinguisher* tersebut memiliki panjang 7,5 putaran. Sifat integral hasil keluaran *AddConstants* dan *AddRoundTweakey* sama dengan keluaran dari *SubCells* sehingga untuk mengefektifkan penulisan, hasil keluaran ketiga fungsi putaran tersebut digabungkan dalam satu matriks yang sama. Proses pencarian *integral distinguisher* terbaik dilanjutkan dengan mencoba seluruh kemungkinan posisi satu, dua, tiga, dan empat sel aktif dengan tahapan pencarian sama seperti yang dilakukan pada satu sel aktif.

4.2. Hasil Pencarian *Integral Distinguisher*

Pencarian *integral distinguisher* pada algoritme SKINNY-64-64 untuk seluruh kemungkinan posisi satu hingga empat sel aktif menghasilkan 2516 *integral distinguisher*. Rekapitulasi hasil pencarian diberikan pada Tabel 4.

Tabel 4 Rekapitulasi hasil pencarian *integral distinguisher*

Panjang putaran	Jumlah *Integral Distinguisher* Berdasarkan Sel Aktif				Jumlah
	Satu Sel	Dua Sel	Tiga Sel	Empat Sel	
5	4	6	4	1	15
5,5	4	16	24	16	60
6	4	40	154	305	503
6,5	4	52	282	856	1194
7	-	6	92	586	684
7,5	-	-	4	56	60
Jumlah	16	120	560	1820	2516

Integral distinguisher yang didapatkan dikelompokkan menjadi enam jenis *integral distinguisher* yaitu *integral distinguisher* 5 putaran, 5,5 putaran, 6 putaran, 6,5 putaran, 7 putaran, dan 7,5 putaran. berdasarkan panjang putaran. Himpunan teks terang dengan satu sel aktif menghasilkan *integral distinguisher* terbaik hingga 6,5 putaran, sedangkan apabila menggunakan dua sel aktif menghasilkan *integral distinguisher* terbaik hingga putaran ketujuh. Himpunan teks terang dengan tiga dan empat sel aktif menghasilkan *integral distinguisher* terbaik dengan panjang yang sama yaitu 7,5 putaran.

Integral distinguisher dengan putaran terpanjang yaitu *integral distinguisher* 7,5 putaran. Pada *integral distinguisher* 7,5 putaran seluruh sel bersifat *U* setelah melewati *MixColumns* putaran kedelapan, sedangkan dalam satu putaran algoritme SKINNY-64-64 fungsi putaran *MixColumns* terdapat setelah melewati fungsi putaran *AddRoundTweakey* yaitu fungsi yang menentukan sel *subtweakey* yang dapat diekstraksi pada setiap putarannya. Oleh karena itu, pada desain serangan menggunakan *integral distinguisher* 7,5 putaran juga menargetkan sel dengan sifat integral *B* pada hasil keluaran *MixColumns* putaran ketujuh untuk melakukan perluasan putaran. Pada *integral distinguisher* 7 putaran seluruh sel bersifat *U* setelah melewati *SubCells* putaran kedelapan sehingga perluasan serangan dilakukan dengan menargetkan sel bersifat *B* pada keluaran *MixColumn* putaran ketujuh. Oleh sebab itu, skenario perluasan putaran serangan menggunakan kedua jenis *integral distinguisher* tersebut dimulai pada putaran

kedelapan. Berdasarkan hal tersebut, dapat disimpulkan bahwa *integral distinguisher* terbaik adalah *integral distinguisher* 7 dan 7,5 putaran.

4.3. Kompleksitas Serangan Menggunakan *Integral Distinguisher* Terbaik

Integral distinguisher 7 dan 7,5 putaran selanjutnya dapat digunakan untuk melakukan ekstraksi *tweakey* pada algoritme SKINNY-64-64. Setiap *integral distinguisher* terbaik yang ditemukan dapat dibuat desain skenario serangan. Desain skenario serangan dimulai dengan menentukan satu sel target untuk melakukan perluasan putaran serangan. Sel target dipilih dari sel yang masih bersifat integral *B* pada hasil keluaran *MixColumns* putaran ketujuh pada *integral distinguisher* 7 dan 7,5 putaran.

Setelah menentukan posisi sel yang dijadikan target, selanjutnya melakukan perluasan serangan dengan melanjutkan proses enkripsi dan memerhatikan posisi-posisi sel yang terlibat untuk menghasilkan sel target ketika dilakukan deskripsi pada proses ekstraksi *tweakey*. Perluasan dilakukan hingga sebelum seluruh enam belas sel dilibatkan untuk mengekstraksi sel target. Apabila *integral distinguisher* 7 dan 7,5 putaran tersebut memiliki lebih dari satu sel yang bersifat *B*, maka proses perluasan serangan untuk setiap *integral distinguisher* tersebut diulang dengan menargetkan sel bersifat *B* pada posisi yang lain.

Kompleksitas dari serangan pada penelitian ini terdiri dari kompleksitas data, memori, dan waktu. Kompleksitas data merupakan banyaknya teks terang terpilih yang dibutuhkan untuk membentuk *integral distinguisher*. Kompleksitas waktu dari serangan ini adalah hasil perkalian dari banyaknya teks terang yang digunakan dan semua kemungkinan nilai dari masukan *tweakey* yang berhasil ditebak atau dapat dituliskan dengan $2^{4d}.2^{4w}$ untuk *d* adalah jumlah sel aktif pada putaran pertama dan *w* merupakan jumlah sel masukan *tweakey* yang berhasil ditebak. Jumlah sel masukan *tweakey* yang ditebak didapatkan dari perluasan serangan.

Pada desain skenario serangan dapat diketahui sel-sel yang terlibat dalam proses deskripsi parsial dan sel-sel *subtweakey* yang ditebak. Sel *subtweakey* yang diekstraksi pada putaran perluasan dapat digunakan untuk mengekstraksi sel masukan *tweakey* menggunakan invers penjadwalan *tweakey* algoritme SKINNY-64-64. Desain skenario serangan yang terbaik menghasilkan kompleksitas waktu terkecil, tetapi dapat mengekstraksi lebih banyak masukan *tweakey*.

Gambar 8 menunjukkan skenario kriptanalisis integral pada SKINNY-64-64 menggunakan *integral distinguisher* 7,5 putaran dengan tiga sel aktif pada posisi (4,11,14) sebagai masukan putaran pertama dan menjadikan keluaran *MixColumns* putaran ketujuh pada posisi (13) sebagai sel target.

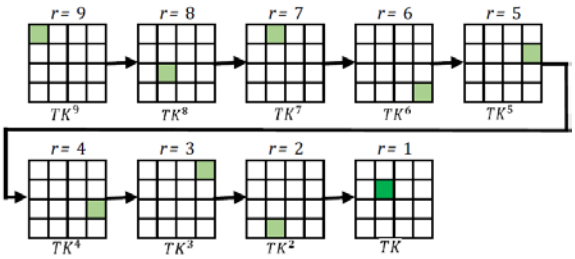
Gambar 8 Desain skenario serangan menggunakan integral distinguisher 7,5 putaran dengan sel aktif pada posisi (4,11,14) dan menargetkan sel pada posisi (13)

Sel berwarna kuning pada Gambar 8. merupakan sel yang terlibat dalam proses ekstraksi *tweakey*. Perluasan serangan dilakukan hingga pada putaran ke-12. Oleh sebab itu, proses enkripsi dilakukan hingga pada putaran ke-12 dan menghasilkan teks sandi yang berkorespondensi sebanyak 2^{12}. Kemudian, proses dilanjutkan dengan melakukan dekripsi secara parsial dengan menghitung invers *MixColumns*. Proses dekripsi dilanjutkan dengan fungsi *ShiftRows* dan *AddRoundTweakey*.

Pada *AddRoundTweakey* dilakukan operasi XOR menggunakan semua kemungkinan nilai *subtweakey* yaitu 2^{12} untuk putaran ke-12. Proses tersebut berlangsung hingga pada putaran posisi sel target yaitu pada putaran kedelapan. Skenario serangan ini seperti yang ditunjukkan pada sel berwarna hijau, dapat mengekstraksi satu sel TK^9 pada posisi (0), dua sel TK^{10} pada posisi (3,4), tiga sel TK^{11} pada posisi (2,5,7), dan lima sel TK^{12} pada posisi (0,1,4,6,7).

Masing-masing *subtweakey* tersebut kemudian dilakukan proses invers penjadwalan *tweakey* pada SKINNY-64-64 yaitu dengan menggunakan invers fungsi permutasi P_T. Proses ini dilakukan untuk mengetahui sel masukan *tweakey* yang dapat diekstraksi. Pada Gambar 9. ditunjukkan proses invers penjadwalan *tweakey* pada TK^9.

Invers penjadwalan *tweakey* dilakukan sampai pada putaran pertama sehingga dari satu sel posisi (0) yang diekstraksi pada TK^9 menghasilkan satu sel masukan *tweakey* TK yang diekstraksi pada posisi (5). Invers penjadwalan *tweakey* untuk TK^{10}, TK^{11}, dan TK^{12} dilakukan sama seperti pada TK^9. TK^9, TK^{10}, TK^{11}, dan TK^{12} dapat mengekstraksi delapan sel masukan *tweakey* pada posisi (1,2,5,8,10,11,12,13).



Gambar 9 Perpindahan posisi sel *subtweakey* yang diekstraksi disebabkan invers penjadwalan *tweakey* pada TK^9 hingga pada putaran pertama

Skenario serangan dengan sel target posisi (13) ini memiliki kompleksitas waktu $2^{12}.2^{32} = 2^{44}$ dan membutuhkan 2^{12} akses memori untuk menyimpan teks sandi. Proses ekstraksi delapan sel *tweakey* lain yang belum diekstraksi dengan menggunakan serangan *brute force* membutuhkan kompleksitas waktu $2^{4.8} = 2^{32}$. Oleh sebab, itu kompleksitas waktu untuk dapat mengekstraksi seluruh sel masukan *tweakey* adalah $2^{44} + 2^{32}$. Proses perluasan serangan dengan *integral distinguisher* pada Gambar 7. menggunakan semua kemungkinan sel target menghasilkan desain skenario terbaik seperti yang ditunjukkan pada Gambar 8. karena posisi sel target lain memiliki kompleksitas total yang lebih besar.

Setelah mencoba melakukan perluasan serangan menggunakan semua *integral distinguisher* 7 dan 7,5 putaran dihasilkan kompleksitas serangan yang berbeda-beda setiap skenario serangan. Tabel 5 menunjukkan hasil rekapitulasi kompleksitas total dari perhitungan kompleksitas menggunakan seluruh *integral distinguisher* 7 dan 7,5 putaran. Perhitungan kompleksitas dilakukan secara teoritis berdasarkan desain skenario kriptanalisis integral pada SKINNY-64-64.

Tabel 5 Rekapitulasi perhitungan kompleksitas total dari desainserangan menggunakan *integral distinguisher* 7 dan 7,5 putaran

Kompleksitas Waktu Total	**Jumlah *Integral Distinguisher* 7 dan 7,5 Putaran**
$2^{44} + 2^{32}$	4
$2^{48} + 2^{24}$	84
$2^{48} + 2^{32}$	55
$2^{52} + 2^{20}$	14
$2^{56} + 2^{24}$	506
$2^{60} + 2^{20}$	81
Jumlah	744

Berdasarkan keenam kompleksitas waktu serangan yang besarnya berbeda tersebut, dapat disimpulkan bahwa kompleksitas waktu yang paling efektif adalah $2^{44} + 2^{32}$. Desain skenario serangan dengan kompleksitas waktu paling efektif dihasilkan dari empat *integral distinguisher* terbaik. Keempat *integral distinguisher* tersebut dibentuk dari himpunan teks terang yang memiliki tiga sel aktif yaitu pada posisi sel $(4,11,14), (5,8,15), (6,9,12),$ dan $(7,10,13)$. Skenario serangan terbaik tersebut mengekstraksi delapan sel *tweakey* dengan menggunakan kriptanalisis integral dan sel *tweakey* lain ditebak menggunakan serangan *brute force*.

5. SIMPULAN DAN SARAN

Berdasarkan hasil penelitian, dapat disimpulkan beberapa hal mengenai penerapan kriptanalisis integral pada algoritme SKINNY-64-64 putaran tereduksi. *Integral distinguisher* terbaik dari hasil yang ditemukan adalah *integral distinguisher* 7 dan 7,5 putaran sebanyak 744. Kedua jenis *integral distinguisher* tersebut dihasilkan dari himpunan teks terang dengan tiga dan empat sel aktif. Kriptanalisis integral pada SKINNY-64-64 menggunakan *integral distinguisher* terbaik memiliki kompleksitas data, memori, dan waktu paling efisien yaitu 2^{12}, 2^{12}, dan 2^{44} secara berturut-turut untuk mengekstraksi delapan sel masukan *tweakey*. Kompleksitas waktu total untuk mengekstraksi seluruh sel masukan *tweakey* adalah sebesar $2^{44} + 2^{32}$. Skenario serangan tersebut menyerang dua belas putaran algoritme SKINNY-64-64 menggunakan empat *integral distinguisher* 7,5 putaran terbaik dengan tiga sel aktif pada posisi $(4,11,14), (5,8,15), (6,9,12),$ dan $(7,10,13)$ pada putaran pertama.

Saran untuk pengembangan selanjutnya adalah perlu dilakukan penelitian lebih lanjut terkait pencarian *integral distinguisher* dengan proses *backward direction* menggunakan metode Nakahara *et al.* [10].

Referensi

[1] W. Stallings, *Cryptography and Network Security Principles and Practice*, Seventh Ed. Edinburgh Gate: Pearson Education, 2017.

[2] M. Liskov, R. L. Rivest, dan D. Wagner, "Tweakable Block Ciphers," in *Proceedings of the 22Nd Annual International Cryptology Conference on Advances in Cryptology*, 2002, hal. 31–46.

[3] C. Beierle *et al.*, "The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS," in *Proceedings, Part II, of the 36th Annual International Cryptology Conference on Advances in Cryptology --- CRYPTO 2016 - Volume 9815*, 2016, hal. 123–153.

[4] B. Ray *et al.*, "The simon and speck families of lightweight block ciphers," *Cryptol. ePrint Arch.*, vol. 2013, no. National Security Agency. USA, hal. 1–42, 2013.

[5] J. Jean, I. Nikolić, dan T. Peyrin, "Tweaks and Keys for Block Ciphers: The TWEAKEY Framework," hal. 274–288, 2014.

[6] L. Knudsen, D. Wagner, C. Berkeley, dan S. Hall, "Integral cryptanalysis ∗ NES/DOC/UIB/WP5/015/1," vol. 1, hal. 1–19, 2002.

[7] M. R. Z'aba, H. Raddum, M. Henricksen, dan E. Dawson, "Bit-pattern based integral attack," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 5086 LNCS, hal. 363–381, 2008.

[8] H. Zhang, W. Wu, dan Y. Wang, "Integral Attack Against Bit-Oriented Block Ciphers," 2016, vol. 9558, hal. 102–118.

[9] Y. Todo, "Structural evaluation by generalized integral property," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9056, hal. 287–314, 2015.

[10] J. Nakahara Jr, P. S. L. . Barreto, B. Preneel, J. Vandewalle, dan H. Y. Kim, "SQUARE Attacks on Reduced-Round PES and IDEA Block Ciphers," 23rd Symp. Inf. Theory Benelux date, vol. 12324, hal. 187–195, 2001.