Jurnal Teknik Informatika (JUTIF)
Vol. 3, No. 4, Agustus 2022, hlm. 897-906

DOI: https://doi.org/10.20884/1.jutif.2022.3.4.352
p-ISSN: 2723-3863
e-ISSN: 2723-3871

MODIFICATION ADVANCED ENCRYPTION STANDARD (AES) ALGORITHM WITH PERFECT STRICT AVALANCHE CRITERION S-BOX

Novita Angraini[*1], Yohan Suryanto[2]

[1,2]Departemen Teknik Elektro Universitas Indonesia, Indonesia
Email: [1]novita.angraini@ui.ac.id, [2]yohansuryanto@ui.ac.id

(Naskah masuk: 31 Mei 2022, Revisi : 07 Juni 2022, diterbitkan: 20 Agustus 2022)

Abstract

The Advanced Encryption Standard or better known as the AES Algorithm is a standard algorithm and has been widely used as an application of cryptography. Currently, a lot of research is developing about attacks on the AES algorithm. Therefore, there have been many studies related to modifications to the AES algorithm with the aim of increasing the security of the algorithm and to produce alternatives to encryption algorithms that can be used to secure data. In this study, modifications were made to the AES algorithm by replacing the S-box using the perfect SAC S-box in the SubBytes process. The Perfect SAC S-box has an exact SAC average value of 0.5. The S-box that will be used must have good security strength, therefore the perfect SAC S-box is tested, namely the AC, SAC, BIC, XOR Table Distribution, and LAT Distribution tests. Based on the results of the study, it was found that the perfect SAC S-box had almost the same S-box test results as the AES S-box. Furthermore, after the perfect SAC S-box is applied to the AES algorithm, it is analyzed how the effect of these modifications on the AES algorithm uses randomness testing for the block cipher algorithm, namely the strict avalanche criterion (SAC) test. The results of the AES test with perfect SAC S-box can meet the SAC test since the second round with better results than the original AES algorithm with SAC values of 0.5003 and 0.5019.

Keywords: *Advanced Encryption Standard (AES), Perfect SAC s-box, S-box, Strict Avalanche Criterion (SAC).*

MODIFIKASI ALGORITME *ADVANCED ENCRYPTION STANDARD* (AES) DENGAN *PERFECT STRICT AVALANCHE CRITERION S-BOX*

Abstrak

Advanced Encryption Standard atau yang lebih dikenal dengan nama Algoritme AES adalah suatu standar algoritme dan telah digunakan secara luas sebagai penerapan dari kriptografi. Saat ini, banyak berkembang penelitian tentang serangan pada algoritme AES. Oleh karena itu telah banyak pula penelitian terkait modifikasi pada algoritme AES dengan tujuan untuk meningkatkan keamanan pada algoritme tersebut serta untuk menghasilkan alternatif dari algoritme enkripsi yang dapat digunakan untuk mengamankan data. Pada penelitian ini, dilakukan modifikasi terhadap algoritme AES dengan mengganti *S-box* menggunakan *perfect* SAC *S-box* pada proses *SubBytes*. Perfect SAC *S-box* memiliki nilai rata-rata SAC yang tepat 0,5. *S-box* yang akan digunakan harus memiliki kekuatan keamanan yang baik, oleh karena itu dilakukan pengujian terhadap *perfect* SAC *S-box*, yaitu uji AC, SAC, BIC, XOR Table Distribution, dan LAT Distribution. Berdasarkan hasil penelitian didapatkan bahwa *perfect* SAC *S-box* memiliki hasil uji *S-box* yang hampir sama dari *S-box* AES. Selanjutnya setelah *perfect* SAC *S-box* diterapkan pada algoritme AES, dianalisis bagaimana pengaruh modifikasi tersebut terhadap algoritme AES menggunakan randomness testing untuk algoritme *block cipher*, yaitu.uji *strict avalanche criterion* (SAC). Hasil uji AES dengan perfect SAC *S-box* dapat memenuhi uji SAC sejak round kedua dengan hasil yang lebih baik dari algoritme AES asli dengan nilai SAC 0,5003 dan 0,5019.

Kata kunci: *Advanced Encryption Standard (AES), Perfect SAC s-box, S-box, Strict Avalanche Criterion (SAC).*

1. PENDAHULUAN

Perkembangan teknologi komunikasi atau sistem informasi telah berkembang secara pesat. Seiring dengan itu, muncul juga risiko ancaman dan tantangan dalam mengamankan system komunikasi dan informasi tersebut, maka dibutuhkan metode keamanan untuk mengantisipasi ancaman tersebut. Sejalan dengan fenomena tersebut, dibutuhkan teknik untuk menjaga kerahasiaan data untuk diterapkan pada teknologi komunikasi dan sistem informasi. Hal ini dapat diatasi dengan menggunakan bagian dari kriptografi, yaitu teknik enkripsi data.

Sistem kriptografi yang sering digunakan untuk enkripsi data adalah sistem kriptografi simetrik block cipher. Block cipher adalah sistem kriptografi simetrik yang memiliki panjang kunci dan panjang blok yang tetap [1]. Salah satu algoritme enkripsi simetrik berbasis block cipher adalah Algoritme Advanced Encryption Standard (AES). Algoritme AES merupakan algoritme standar berdasarkan FIPS Publication, 2001.

Advanced Encryption Standard atau yang lebih dikenal dengan nama Algoritme AES adalah suatu standar algoritme enkripsi simetrik berbasis block cipher yang telah digunakan secara luas sebagai penerapan dari kriptografi. Algoritme AES terdiri dari empat komponen utama, yaitu SubBytes dengan Substitution Box (S-box), Shiftrows, Mixcolumn, dan Addroundkey [1]. Pada umumnya algoritme block cipher harus memenuhi konsep difusi dan konfusi. Berdasarkan penjelasan Stallings dalam buku Cryptography and Network Security [2], difusi adalah menyebarkan struktur statistik plaintext ke dalam struktur statistik yang melibatkan kombinasi yang panjang dari bit-bit dalam ciphertext. Atau dapat diartikan bahwa difusi adalah menyebarkan pengaruh dari satu bit input ke sebanyak mungkin bit output. Difusi dapat dilakukan melalui permutasi Dalam AES, difusi dapat dilakukan melalui Shiftrows dan Mixcolumn. Sedangkan pengertian konfusi adalah penggunaan transformasi penyandian untuk mempersulit pencarian hubungan statistik antara input dan output. Salah satu komponen dalam Algoritme AES yang menggunakan prinsip konfusi adalah S-box. S-box merupakan salah satu komponen nonlinear dari suatu algoritme. S-box yamg memiliki properti kriptografis yang baik membuat algoritme enkripsi tahan terhadap serangan [3].

S-box AES akan memetakan 8 bit input menjadi 8 bit output yang merupakan proses substitusi input menjadi output. Proses ini bisa dilakukan dengan perhitungan matematis GF 2^8 atau menggunakan tabel. S-box AES memiliki kekuatan keamanan yang cukup baik. Saat ini banyak berkembang metode pembangkitan S-box. Banyak hasil pembangkitan S-box yang mendekati dengan ciri-ciri dan kekuatan dari S-box AES. Penelitian mengenai pembangkitan S-box telah dilakukan Al-Dweik, et al [4], yaitu pembangkitan Key-Dependent S-boxes menghasilkan S-box yang memiliki sifat aljabar yang sama dengan S-box pada algoritme AES. S-box yang dihasilkan pun telah diuji menggunakan uji S-box dan memiliki hasil uji yang persis sama dengan hasil uji S-box asli AES. Selanjutnya penelitian mengenai pembangkitan S-box telah dilakukan Musheer Ahmad, et al [5]. Pada penelitian ini, pembangkitan S-box menggunakan pengembangan Chaotic Map berdasarkan pencarian heuristic dan struktur grup aljabar. S-box yang dihasilkan memiliki hasil uji yang mendekati dengan hasil uji S-box asli AES. Penelitian lainnya telah dilakukan oleh Alamsyah [6], yaitu pembangkitan S-box menggunakan 17 buah irreducible polynomial dan transformasi affine dengan tujuan mendapatkan perfect SAC S-box, yaitu S-box yang memiliki nilai uji SAC tepat 0,5.

Perkembangan dari penggunaan S-box yang lain pada proses SubBytes juga telah banyak dilakukan karena S-box merupakan salah satu komponen utama yang dapat memenuhi sifat konfusi pada algoritme AES. Modifikasi algoritme AES menggunakan pada S-box telah dilakukan oleh Thin dan Thwin [7] yang menamakan algoritme modifikasinya sebagai AES-R, pada AES-R ditambahkan second key dan modifikasi SubBytes yang dinamakan TransportSubBytes. Hasil yang didapatkan adalah algoritme AES-R memiliki performa enkripsi yang lebih cepat dari AES dan memiliki avalanche effect dan entropi yang hampir sama dengan algoritme AES. Penelitian lain mengenai modifikasi algoritme AES dengan penggunaan S-box yang lain pada proses SubBytes telah dilakukan oleh Chauhan, et al [8], pada skema tersebut digunakan Key Dependent Dynamic S-box untuk menggantikan S-box static AES dengan tujuan meningkatkan keamanan pada algoritme AES. Hasil yang didapatkan, dengan modifikasi AES menggunakan Key Dependent Dynamic S-box memiliki avalanche Effect yang lebih baik dari pada algoritme asli AES. Penelitian lainnya mengenai modifikasi algoritme AES telah dilakukan oleh Gamido, et al [9], pada skema yang diajukan, modifikasi algoritme AES menggunakan bit permutation untuk mengganti matriks pada proses Mixcolumn dengan tujuan untuk mempermudah implementasi algoritme AES dan menyerhanakan komputasi matematika pada algoritme tersebut. Hasil yang didapatkan, dengan modifikasi AES menggunakan bit permutation memiliki avalanche effect yang lebih baik dan meningkatkan performa dari pada algoritme asli AES.

Pada penelitian ini, dilakukan modifikasi terhadap algoritme AES dengan mengganti S-box pada proses SubBytes. S-box yang akan digunakan adalah S-box yang dibangkitkan oleh Alamsyah [6] yang memiliki nilai rata-rata SAC sebesar 0,5 sehingga S-box tersebut dikatakan perfect SAC S-box. Namun dalam penelitian tersebut, perfect SAC S-box, hanya dilakukan pengujian SAC. Oleh karena itu, sebelum diterapkan pada algoritme AES, perlu dilakukan pengujian S-box lainnya agar dapat diketahui kekuatan kriptografis dari S-box tersebut. Setelah itu S-box tersebut akan diterapkan pada proses subBytes dari algoritme AES yang merupakan algoritme standar yang digunakan saat ini. Selanjutnya untuk mengetahui dan menganalisis pemenuhan sifat konfusi dan difusi dari algoritme modifikasi tersebut, maka dilakukan pengujian menggunakan randomness testing untuk algoritme block cipher, yaitu.uji strict avalanche criterion (SAC) terhadap algoritme hasil modifikasi AES tersebut.

2. METODE PENELITIAN

Tahapan-tahapan yang dilakukan pada penelitian ini adalah sebagai berikut:

1. Pengujian terhadap *perfect* SAC *S-box* menggunakan uji AC, SAC, BIC, XOR *Table Distribution*, dan LAT *Distribution*.
2. Analisis kekuatan kriptografis dari *perfect* SAC *S-box* berdasarkan hasil pengujian yang telah dilakukan.
3. Penerapan *perfect* SAC *S-box* pada algoritme AES..
4. Pengujian algoritme AES yang menggunakan *perfect* SAC *S-box*. Pengujian yang digunakan adalah uji SAC terhadap keseluruhan algoritme. Variabel yang dibutuhkan dalam uji SAC meliputi variabel bebas, variabel kontrol dan variabel terikat. Variabel uji SAC yang digunakan dalam penelitian ini dapat dilihat pada Tabel 1.
5. Analisis hasil pengujian dilakukan berdasarkan hasil uji SAC dari algoritme AES yang menggunakan *perfect* SAC *S-box.*
6. Penarikan simpulan mengenai implementasi *perfect* SAC *S-box* algoritme AES.

Tabel 1. Variabel Uji

Pengujian	Objek Pengujian	Variabel		
		Input		Output Terkait
		Bebas	Kontrol	
SAC	Algoritma modifikasi AES	Kunci	*Plaintext*	*Cipher text*
		Plaintext	Kunci	*Cipher text*
AC	*perfect* SAC *S-box*	-	-	-
BIC	*perfect* SAC *S-box*	-	-	-
LAT	*perfect* SAC *S-box*	-	-	-
XOR Table	*perfect* SAC *S-box*	-	-	-

2.1. Pengujian S-Box

Keamanan algoritma enkripsi *block cipher* salah satunya tergantung pada *S-box* yang digunakan pada fungsi *round*. *S-box* merupakan salah satu komponen nonlinear dari suatu algoritma. Sebuah *S-box S* berukuran *n* x *m* dengan *n input* dan *m output* adalah sebuah pemetaan $S: \{0,1\}^n \rightarrow \{0,1\}^n$.

Pengujian terhadap *S-box* menggunakan uji AC, SAC, BIC, XOR *Table Distribution*, dan *Linear Approximation Table* (LAT). Pada penelitian ini pengujian *S-box* dinyatakan memenuhi atau lebih baik untuk seluruh kriteria uji saat dibandingkan dengan hasil uji *S-box* AES.

a. Uji *Avalanche Criterion* (AC)

Uji AC berkaitan dengan pengaruh dari perubahan bit *input* terhadap perubahan bit *output*. Kondisi ideal dari suatu *S-box* apabila setiap satu *input* bit nilainya diubah, maka setengah dari bit ouput harus berubah [11]. Berdasarkan penjelasan Yucel dan Vergili [12], Sebuah fungsi $f: \{0,1\}^n \rightarrow \{0,1\}^n$ dikatakan memenuhi *avalanche criterion* jika ketika satu bit *input* diubah dapat mempengaruhi perubahan rata-rata setengah dari bit *output*, dimana *i* dan $j \in (1,2,\dots,n)$ adalah bit *input* dan *output*. Persamaan matematis dari definisi tersebut adalah:

$$\frac{1}{2^n}\sum_{j=1}^{n} W\left(a_j^{e_i}\right) = \frac{n}{2} \quad (1)$$

Keterangan: e_i adalah vektor *unit* bit ke-*i* yang bernilai 1 dan bit yang lain bernilai 0.

$a_j^{e_i}$ adalah variabel *avalanche* ke-*j*

$$W\left(a_j^{e_i}\right) = \sum_{all\ X \in \{0,1\}^n} \left(a_j^{e_i}\right) \quad (2)$$

adalah total perubahan pada variabel *avalanche* ke-*j* terhadap seluruh *input* berukuran 2^n dimana $0 \leq W\left(a_j^{e_i}\right) \leq 2^n$.

Dengan menggunakan persamaan (1) dapat dilakukan modifikasi untuk menentukan parameter *avalanche*, $k_{AVAL}(i)$ sebagai berikut:

$$k_{AVAL}(i) = \frac{1}{n2^n}\sum_{j=1}^{n} W\left(a_j^{e_i}\right) = \frac{1}{2} \quad (3)$$

$k_{AVAL}(i)$ berada pada rentang [0,1] dan merupakan probabilitas perubahan dari seluruh bit *output* ketika bit *input* ke-*i* diubah.

Pada setiap pengujian, karena *S-box* yang diuji memetakan 8 bit *input* menjadi 8 bit *output*, maka terdapat *input* dari *S-box* sebanyak $\mathbf{2^8}$ dan nilai *vector unit*. Nilai *vector unit* adalah sejumlah bit dengan panjang bit sama dengan panjang bit *input* dan memiliki bobot 1 untuk mengubah satu bit *input* ke seluruh posisi bit *output*. Setiap kemungkinan *input* akan masuk ke dalam *S-box* dan menghasilkan nilai *output* dari *S-box*. Nilai *vector unit* digunakan untuk mengubah kemungkinan *input* yang selanjutnya akan dimasukkan ke dalam fungsi *S-box* dan menghasilkan *output* dari *S-box* tersebut. Selanjutnya, masing-masing *output S-box* tersebut akan dihitung nilai *avalanche vector*-nya berdasarkan nilai *input S-box* tersebut. Nilai *avalanche vector* merupakan hasil XOR antara *output S-box* dari setiap kemungkinan *input* dengan *output S-box* dari setiap kemungkinan *input* yang telah diubah oleh *vector unit*. Kemudian dihitung nilai $\boldsymbol{k_{AVAL}}$ menggunakan persamaan (3) dan dihitung nilai errornya ($\boldsymbol{k_{AVAL}} - 0{,}5$).

b. Uji *Strict Avalanche Criterion* (SAC)

SAC merupakan peningkatan dari AC, dimana saat bit *input* berubah, dapat mengubah bit *output* dengan probabilitas tepat ½. Menurut Yucel dan Vergili [12], sebuah fungsi $f: \{0,1\}^n \rightarrow \{0,1\}^n$ dikatakan memenuhi SAC jika untuk semua *i* dan $j \in (1,2,\ n)$, saat bit *input* ke-*i* berubah, maka akan mengubah bit *output* ke-*j* dengan probabilitas tepat ½. Formulasi dari penjelasan tersebut menjadi:

$$\frac{1}{2^n} W\left(a_j^{e_i}\right) = \frac{1}{2} \quad (4)$$

Persamaan di atas dapat disusun untuk menentukan parameter SAC, $k_{SAC}(i,j)$ yaitu:

$$k_{SAC}(i,j) = \frac{1}{2^n} W\left(a_j^{e_i}\right) = \frac{1}{2} \quad (5)$$

$k_{SAC}(i,j)$ berada pada rentang [0,1] dan merupakan probabilitas perubahan bit *output* ke-*j* ketika terjadi perubahan pada bit *input* ke-*i*. Suatu *S-box* memenuhi uji SAC apabila semua nilai ketergantungan idealnya mendekati setengah [13].

Dalam pengujian SAC pada *S-box*, *output avalanche vector* akan dihitung nilai *weight* (bobot) pada setiap posisi bit *output S-box* terhadap seluruh kemungkinan nilai *input*. Kemudian dari hasil perhitungan nilai *weight* tiap posisi *output avalanche vector* tersebut, akan dilihat apakah jumlah *weight* dibandingkan dengan jumlah bit *output* ke-*j* tersebut memenuhi atau mendekati kriteria untuk *Strict Avalanche Criterion*. Perhitungan $\boldsymbol{k_{SAC}(i,j)}$ dilakukan dengan menggunakan persamaan (5) dan dihitung nilai errornya ($\boldsymbol{k_{SAC}} - 0{,}5$).

c. Uji *Linear Approximation Table* (LAT)

Menurut [14], pada pengujian LAT Distribution, akan dilakukan perhitungan nilai LAT yaitu membandingkan nilai xor dari bit *input* yang dioperasikan dengan α dengan nilai xor dari bit *output* yang dioperasikan dengan β. Dengan demikian diketahui berapa banyak *output S-box* yang merupakan fungsi linear. Setelah dilakukan perhitungan nilai LAT akan dihitung nilai bias pada setiap nilai LAT, dimana nilai bias adalah hasil probabilitas dikurangi dengan nilai 1/2. Nilai bias pada LAT diharapkan merata dengan jumlah yang sama.

Nilai LAT ideal untuk *S-box* ukuran 8x8 adalah sebesar 128. Dengan demikian probabilitas nilai LAT-nya adalah 128/256 dan nilai biasnya adalah 0. Semakin kecil nilai bias dalam *linear approximation*, maka semakin tahan terhadap kriptanalisis linear [15]. Hal ini menunjukkan tidak adanya korelasi *input* dengan *output*. Jika nilai probabilitas *S-box* kurang atau lebih dari ½ maka terdapat korelasi antara *input* dengan *output*. Korelasi tersebut dapat dimanfaatkan untuk membentuk persamaan linear *S-box*. Hal ini dapat menyebabkan adanya linear cryptanalysis.

d. Uji Bit Independence Criterion (BIC)

Sebuah fungsi $f: \{0,1\}^n \rightarrow \{0,1\}^n$ dikatakan memenuhi BIC jika untuk semua *i*, *j*, *k* ∈ {1,2, … , n} dengan $j \neq k$, perubahan bit *input* ke-*i* menyebabkan bit *output* ke-*j* dan ke-*k* berubah secara bebas.

Untuk mengukur nilai bit *independence* dibutuhkan nilai koefisien korelasi antara komponen ke-*j* dan ke-*k* dari *difference output* yang disebut dengan *avalanche vector* A^{e_i}. Parameter BIC berhubungan dengan pengaruh perubahan bit *input* ke-*i* pada bit *output* ke-*j* dan ke-*k* dari A^{e_i}, yang didefinisikan dengan:

$$BIC(a\_j, a\_k) = \max_{1 \le i \le n} \left| corr\left(a_j^{e_i}, a_k^{e_i'}\right) \right| \quad (6)$$

Secara keseluruhan parameter BIC ditetapkan sebagai

$$BIC(f) = \max_{1 \le j,k \le n, j \neq k} BIC(a_j, a_k) \quad (7)$$

Nilai $BIC(f)$ berada pada rentang [0,1] dan memiliki kriteria sebagai berikut:

- Jika nilai $BIC(f) = 1$ maka *avalanche variable*-nya identik atau kebalikannya.
- Jika nilai $BIC(f) = 0$ maka *avalanche variable*-nya saling bebas

e. Uji XOR *Table Distribution*

Pengujian XOR *Table Distribution* untuk suatu *S-box*, akan dilihat perbedaan *output S-box* dengan tiap kemungkinan *input* dan *output. S-box* dengan *input* yang telah di-XOR dengan *vector unit*. Perubahan *output S-box* yang disebabkan oleh vektor *unit* e akan dihitung dengan rumus yaitu $b = f(P) \oplus f(P \oplus \delta)$. Hasil perubahan *output S-box* b tersebut akan diubah menjadi suatu bilangan intejer dan jumlahnya akan disimpan sebagai entri XOR *Table*. Untuk pengujian XOR *Table* akan dicari nilai tertinggi dari entri XOR *Table*, dimana jumlah entri selalu genap dan jumlah nilai pada barisnya adalah 2^n.

Kriteria uji XOR *Table* berkaitan dengan ketahanan *S-box* terhadap *differential cryptanalysis.* Nilai entri tertinggi pada XOR *Table* menunjukkan probabilitas penerapan *differential cryptanalysis* pada *S-box* tersebut, dan jumlah dari nilai entri XOR *Table* menunjukkan jumlah kemungkinan dibentuknya pasangan *input* dan *output difference* yang dimanfaatkan pada *differential cryptanalysis* tersebut.

2.2. Modifikasi Algoritma AES

Pada tahun 2001, algoritma *Block Cipher* yang terpilih sebagai *Advanced Encryption Standard* (AES) adalah algoritma Rijndael. Algoritma tersebut diajukan oleh Joan Daemen dan Vincent Rijmen. AES adalah algoritma simetrik standar berbasis *block cipher* yang mengenkripsi 128-bit blok *input* menjadi 128-bit blok *output*. Algoritma AES menggunakan beberapa panjang kunci berbeda, yaitu 128, 192, dan 256 bit. Berdasarkan panjang kunci tersebut, AES dikelompokkan menjadi AES-128, AES-192, dan AES-256 [2].

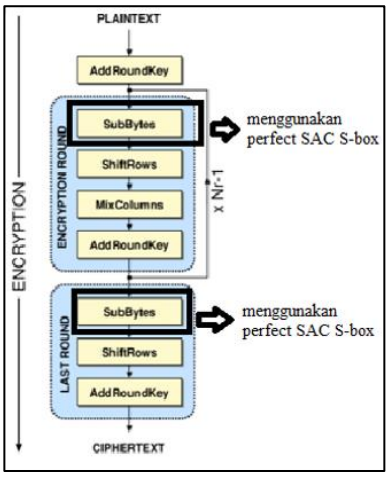
Output ciphertext pada algoritma AES diperoleh dengan melakukan transformasi dari blok *input*. Transformasi dilakukan berdasarkan fungsi *round* sebanyak 10, 12 atau 14 kali (tergantung panjang kunci yang digunakan). Adapun komponen transformasi dalam fungsi *round* sebagai berikut:

1) Substitusi *byte* menggunakan *S-box* (*S-box*).
2) *Shifting rows* dari *state array* (*ShiftRow*)
3) *Mixing data* dalam setiap kolom dari *state array* (*Mixcolumn*)

4) Penambahan *round key* ke. *state* (*AddRoundKey*)

Untuk *round* terakhir terdapat perbedaan dengan *Nr-1 round* pertama yaitu tidak terdapat transformasi *Mixcolumn*().

Ilustrasi dari modifikasi AES dapat dilihat pada gambar 1 di bawah ini.



Gambar 1. Modifikasi AES dengan *Perfect* SAC *S-box*

Pada penelitian ini dimodifikasi algoritma AES menggunakan *perfect* SAC *S-box* yang akan diimplementasikan pada proses *subbytes*. Algoritma AES yang digunakan untuk dimodifikasi algoritma AES dengan kunci 128 bit, yang sering disebut AES 128. Algortima dengan panjang kunci 128 bit juga merupakan panjang kunci yang direkomendasikan oleh BSI [16], ECRYPT [17], dan NIST [18].

2.3. Pengujian Algoritma

Pada penelitian Kavut dan Yucel [19], telah dilakukan pengujian properti keamanan, baik untuk *S-box* dan juga untuk algoritma keseluruhan pada algoritma Rijndael yang merupakan algoritma AES. Pada penelitian ini, setelah menerapkan modifikasi pada algoritma AES dengan mengganti *S-box* pada proses SubBytes, maka akan dilakukan pengujian terhadap algoritma modifikasi tersebut. Pada umumnya untuk melihat keacakan dari suatu algoritma block cipher adalah dengan melihat *avalanche effect*nya.

Berdasarkan [19], *avalanche effect* adalah seberapa banyak jumlah bit yang berubah pada *output ciphertext* (teks sandi) saat satu bit *input plaintext* (teks terang) diubah. Hal ini menunjukkan tingkat difusi dari suatu algoritma block cipher. Selanjutnya dari avalanche effect dapat ditingkatkan menjadi Strict Avalanche Criterion, yaitu saat setiap kali satu bit *input* diubah, setiap bit *output* harus berubah dengan probabilitas tepat setengah untuk mencapai difusi yang ideal. Algoritma AES asli telah dilakukan uji SAC dengan nilai yang cukup baik.

Oleh karena itu, pada algoritma modifikasi AES perlu dilakukan pengujian untuk melihat sifat konfusi dan difusi dari algoritma tersebut. Pada penelitian [20], untuk membuktikan analisis terhadap sifat konfusi dan difusi, dapat dilakukan dengan pengujian SAC menggunakan kunci dan juga plainteks yang dijadikan variabel kontrol yang dibuat tetap secara bergantian. Setelah melakukan pengujian tersebut, hasil uji akan dibandingkan dan dianalisis dengan hasil uji pada algoritma asli AES. Algoritma AES yang digunakan pada penelitian ini adalah Algoritma AES 128 sebanyak 10 *round* yang memiliki *input plaintext* dan kunci sebesar 128 bit, sehingga populasi *plaintext* dan kunci adalah 2^{128}.

Untuk pengujian SAC keseluruhan algoritma AES yang telah diterapkan *perfect* SAC *S-box* dilakukan dalam dua tahap. Tahap pertama adalah pembangkitkan variabel bebas sebanyak 20000 sampel. Ketika *plaintext* menjadi variabel bebas, maka kunci dijadikan sebagai variabel kontrol dengan menggunakan nilai ekstrim nol. Sedangkan ketika kunci menjadi variabel bebas, maka *plaintext* dijadikan sebagai variabel kontrol dengan nilai ekstrim nol. Penggunaan nilai nol pada variabel kontrol dibuat agar variabel kontrol ini tidak mempengaruhi variabel utama, karena yang akan diuji adalah pengaruh variabel bebas terhadap variabel terikat. *Output* dari proses tersebut adalah nilai variabel terikat (*ciphertext*).

Tahap kedua adalah melakukan pengujian SAC dari sampel yang telah dibangkitkan dengan menggunakan perhitungan uji SAC. Sampel pada penelitian ini menggunakan sebanyak 20000 buah sampel. Pengujian SAC ini dilakukan setiap round, dengan jumlah round sebanyak 10 round. Hal ini dilakukan dengan tujuan untuk mengetahui posisi round dari algoritma yang diuji dapat memenuhi SAC dengan baik.

3. HASIL DAN PEMBAHASAN

Perfect SAC *S-box* yang digunakan untuk memodifikasi algoritma AES dibangkitkan dengan cara sebagai berikut:

a. Untuk setiap elemen di *finite field* $GF(2^8)$, ambil *invers* multiplikatif dengan *irreducible polynomial*

$$x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + 1$$

Elemen 0x00 dipetakan ke 0x00.

b. Lakukan transformasi *affine* atas $GF(2)$ pada hasil dari poin a. Elemen transformasi *affine* dari *S-box* yang digunakan dapat dinyatakan dalam bentuk matriks berikut

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Tabel 2. merupakan hasil pembangkitan *perfect* SAC *S-box*

Tabel 2. *Perfect* SAC *S-box*

63	**7C**	**D7**	**44**	**2**	**81**	**F0**	**F3**	**E8**	**13**	**12**	**24**	**91**	**74**	**10**	**C2**
1E	D6	8E	F8	BD	A7	FA	88	D8	64	B1	6C	86	67	EC	21
27	45	FE	9	E2	C3	C6	F	99	CE	A8	26	14	B0	DE	A
5B	5D	34	A6	ED	83	20	4D	F5	8F	79	4C	11	66	2D	E6
65	BB	8	E5	4E	57	F1	FF	CA	48	9A	2A	F9	72	F7	84
75	5	71	D0	E9	2B	5C	5E	18	D2	2C	7D	87	43	AC	37
7A	A9	70	35	AD	7B	6D	32	98	41	33	3	8A	52	55	C9
9D	2E	60	28	E0	F4	FB	6E	1A	DA	D3	61	E1	A1	B3	7F
9E	1F	49	D4	4B	CC	68	69	97	17	C0	A3	78	D1	36	A2
A0	50	E	53	D	BA	C5	6A	4F	47	0	1D	E3	FD	DC	FC
B7	59	CD	22	9F	38	C7	B2	15	3A	EB	EE	29	B8	AB	F2
B9	F6	C1	25	D5	51	40	77	54	7E	B4	9C	B	1B	E7	6B
BE	16	DB	BC	31	92	DF	C4	AA	89	5A	80	A4	B6	42	C8
DD	D9	82	8D	AE	8C	95	3F	C	9B	1	4A	94	8B	96	6
E4	CF	BF	58	3B	A5	62	1C	19	B5	39	46	30	90	56	3C
EF	3E	3D	7	EA	2F	73	93	4	AF	6F	85	5F	76	CB	23

3.1. Hasil Uji AC *S-box*

Berikut ini adalah hasil pengujian AC dari *Perfect* SAC *S-box* yang diterapkan pada algoritma AES. Hasil pengujian disajikan dalam bentuk tabel nilai AC dan nilai error dari uji AC. Tabel 3 adalah hasil uji AC dan nilai error dari *Perfect* SAC *S-box* dan AES *S-box*.

Tabel 3. *Nilai* AC dari *perfect* SAC *S-box* dan AES *S-box*

Posisi bit	**Nilai AC**		**Nilai Error AC**	
	Perfect SAC *S-box*	*S-box* AES	*Perfect* SAC *S-box*	*S-box* AES
1	0,48242	0.49219	0,01758	0,00781
2	0,50781	0.49805	0,00781	0,00195
3	0,49805	0.51172	0,00195	0,01172
4	0,49414	0.50781	0,00586	0,00781
5	0,50000	0.5	0,00000	0
6	0,51367	0.50391	0,01367	0,00391
7	0,51758	0.50781	0,01758	0,00781
8	0,48633	0.51758	0,01367	0,01758
Nilai Max	0,51758	0.51758	0,01758	0,01758

Berdasarkan tabel 3 dapat dilihat bahwa nilai maksimal uji AC dari *perfect* SAC *S-box* memiliki nilai maksimal uji AC yang sama dengan *S-box* AES, yaitu sebesar 0,51758.

3.2. Hasil Uji SAC *S-box*

S-box memenuhi SAC jika perubahan pada setiap satu bit *input* menyebabkan setiap bit *output* pada *S-box* berubah dengan probabilitas mendekati ½ atau nilai error (probabilitas dikurang ½) mendekati 0. Berdasarkan uji SAC yang telah dilakukan, *Perfect* SAC *S-box* memiliki nilai error sebesar 0,0469.

Sedangkan *S-box* AES memiliki nilai error SAC sebesar 0.0625. Tabel 4 dan tabel 5 menunjukkan hasil uji SAC dan nilai error maksimal dari setiap posisi bit *input* pada *Perfect* SAC *S-box*.

Tabel 4. Nilai SAC dari *Perfect* SAC *S-box*

0,5469	0,4531	0,4531	0,4531	0,4688	0,4531	0,5313	0,5
0,5	0,5469	0,5	0,5156	0,5469	0,4688	0,5469	0,4375
0,4375	0,5	0,4844	0,5	0,5156	0,5469	0,5156	0,4844
0,4844	0,4375	0,5	0,4688	0,4844	0,5156	0,5156	0,5469
0,5469	0,4844	0,5313	0,5156	0,4688	0,4844	0,4688	0,5
0,5	0,5469	0,5469	0,4844	0,5156	0,4688	0,5156	0,5313
0,5313	0,5	0,5313	0,5156	0,4688	0,5156	0,5469	0,5313
0,5313	0,5313	0,4688	0,5	0,4688	0,4688	0,4375	0,4844
Nilai rata-rata SAC							**0,5000**

Tabel 5. Nilai Error SAC dari *Perfect* SAC *S-box*

0,0469	0,0469	0,0469	0,0469	0,0312	0,0469	0,0313	0
0	0,0469	0	0,0156	0,0469	0,0312	0,0469	0,0625
0,0625	0	0,0156	0	0,0156	0,0469	0,0156	0,0156
0,0156	0,0625	0	0,0312	0,0156	0,0156	0,0156	0,0469
0,0469	0,0156	0,0313	0,0156	0,0312	0,0156	0,0312	0
0	0,0469	0,0469	0,0156	0,0156	0,0312	0,0156	0,0313
0,0313	0	0,0313	0,0156	0,0312	0,0156	0,0469	0,0313
0,0313	0,0313	0,0312	0	0,0312	0,0312	0,0625	0,0156
Nilai error terkecil							**0,0469**

Tabel 6. dan Tabel 7. adalah hasil uji SAC *S-box* dan nilai error maksimal dari AES *S-box*

Tabel 6. Nilai SAC dari AES *S-box*

0,5156	0,5156	0,4531	0,5625	0,4531	0,4844	0,4531	0,5000
0,4688	0,4844	0,5625	0,5000	0,4844	0,4531	0,5000	0,5313
0,5156	0,5156	0,5000	0,4688	0,5625	0,5000	0,5313	0,5000
0,5313	0,5313	0,4688	0,4531	0,5000	0,5313	0,5000	0,5469
0,4531	0,5000	0,4531	0,5156	0,5000	0,5000	0,5469	0,5313
0,4531	0,5156	0,5156	0,4688	0,4688	0,5469	0,5313	0,5313
0,5313	0,5313	0,4688	0,5156	0,4688	0,5313	0,5313	0,4844
0,5156	0,5625	0,5156	0,5313	0,4844	0,5313	0,4844	0,5156
Nilai rata-rata SAC							**0,50488**

Tabel 7. Nilai Error SAC dari AES *S-box*

0,0156	0,0156	0,0469	0,0625	0,0469	0,0156	0,0469	0,0000
0,0313	0,0156	0,0625	0,0000	0,0156	0,0469	0,0000	0,0313
0,0156	0,0156	0,0000	0,0313	0,0625	0,0000	0,0313	0,0000
0,0313	0,0313	0,0313	0,0469	0,0000	0,0313	0,0000	0,0469
0,0469	0,0000	0,0469	0,0156	0,0000	0,0000	0,0469	0,0313
0,0469	0,0156	0,0156	0,0313	0,0313	0,0469	0,0313	0,0313
0,0313	0,0313	0,0313	0,0156	0,0313	0,0313	0,0313	0,0156
0,0156	0,0625	0,0156	0,0313	0,0156	0,0313	0,0156	0,0156
Nilai Error terkecil							**0,0625**

3.3. Hasil LAT

Nilai dari uji *linear approximation table* (LAT) merupakan parameter dari suatu *S-box* terhadap ketahanan dari *linear attack*. Berdasarkan hasil uji yang dilakukan, nilai dari uji LAT *S-box* yang digunakan pada *Perfect* SAC *S-box* berada di antara rentang nilai -16 dan 16. Rentang nilai ini sama seperti rentang nilai dari uji LAT *S-box* yang digunakan pada algoritme AES. Berdasarkan. [21], fungsi yang memenuhi sifat *Almost Bent* (AB) adalah *S-box* yang nilai LAT hanyalah $\{0, \pm 2^{\frac{n+1}{2}}\}$, tetapi itu hanya dimiliki oleh *S-box* berukuran ganjil. Fungsi AB memberikan resistansi optimal terhadap serangan diferensial dan linier. Untuk n genap, fungsi dengan nonlinier $2^{n-1} - 2^{\frac{n}{2}}$ diketahui dan diperkirakan bahwa nilai ini adalah yang tertinggi dalam hal ini. *Perfect* SAC *S-box* dan *S-box* AES berukuran genap (dengan *input* 8 bit dan *output* 8 bit) dengan nilai LAT terbesarnya adalah $\pm 2^{\frac{n}{2}}$ (-16 dan 16).

Berdasarkan hasil uji LAT, maka nilai bias *Perfect* SAC *S-box* yang dihasilkan adalah ±16/256, yaitu sama dengan nilai bias *S-box* AES yaitu ±16/256.

3.4. Hasil BIC

Berdasarkan hasil uji BIC yang dilakukan terhadap *Perfect* SAC *S-box*, didapatkan hasil nilai BIC maksimal adalah sebesar 0.13498. Jika dibandingkan dengan hasil uji BIC dari *S-box* AES yang memiliki nilai BIC maksimal sebesar 0.13412, maka hasil uji BIC *Perfect* SAC *S-box*, tidak lebih baik dari *S-box* AES.

3.5. Hasil XOR *Table Distribution*

Nilai dari uji *XOR table distribution* merupakan parameter ketahanan dari suatu *S-box* terhadap *differential attack.* Berdasarkan hasil uji yang dilakukan, nilai terbesar dari uji *XOR table distribution S-box* dari *Perfect* SAC *S-box* adalah nilai 4. Nilai ini sama besarnya dengan nilai terbesar dari uji *XOR table distribution S-box* yang digunakan pada algoritme AES.

Tabel 8. adalah sebaran nilai XOR *table* dari *Perfect* SAC *S-box* dan AES *S-box*.

Tabel 8. Nilai XOR *table* dari *Perfect* SAC *S-box* dan AES *S-box*.

S-box	**Jumlah entri XOR**			
	0	2	4	256
AES	33150	32130	255	1
Perfect* SAC *S-box	33150	32130	255	1

3.6. Hasil Uji SAC Algoritma

Berdasarkan hasil uji SAC dengan variabel bebas kunci, algoritma AES dengan *Perfect* SAC *S-box* memiliki nilai rata-rata SAC 0,5003 pada *round* kedua. Sedangkan Algoritma AES asli memiliki nilai rata-rata SAC 0,5004 pada *round* kedua.

Tabel 9. Nilai SAC bebas kunci dari AES *perfect* SAC *S-box*

Round	**Nilai SAC bebas kunci**			
	minimal	maksimal	rata-rata	Error rata-rata
1	0	1	0,1630	0,3370
2	0,48745	0,51725	0,5003	0,0003
3	0,48465	0,5141	0,5000	0,0000
4	0,4852	0,51305	0,5000	0,0000
5	0,4866	0,51415	0,5000	0,0000
6	0,48545	0,5154	0,5000	0,0000
7	0,4858	0,5143	0,5000	0,0000
8	0,48745	0,51295	0,5000	0,0000
9	0,4856	0,51465	0,5000	0,0000
10	0,4851	0,51255	0,5000	0,0000

Tabel 9. dan Tabel 10. merupakan hasil pengujian SAC dengan variabel bebas kunci. Dalam tabel tersebut berisi nilai SAC minimal, maksimal dan rata-rata serta nilai error rata-rata.

Tabel 10. Nilai SAC bebas kunci dari AES

Round	**Nilai SAC bebas kunci**			
	minimal	maksimal	rata-rata	Error rata-rata
1	0	1	0,1649	0,3351
2	0,48635	0,5156	0,5004	0,0004
3	0,4871	0,5139	0,5000	0,0000
4	0,4858	0,5133	0,5000	0,0000
5	0,48755	0,5134	0,5000	0,0000
6	0,48685	0,51305	0,5000	0,0000
7	0,48595	0,51395	0,5000	0,0000
8	0,4865	0,51395	0,5000	0,0000
9	0,4874	0,5133	0,5000	0,0000
10	0,48645	0,5152	0,5000	0,0000

Tabel 11. dan Tabel 12. merupakan hasil pengujian SAC dengan variabel bebas plainteks. Algoritma AES dengan *Perfect* SAC *S-box* memiliki nilai rata-rata SAC 0,5019 pada *round* kedua. Sedangkan Algoritma AES asli memiliki nilai rata-rata SAC 0,5020 pada *round* kedua.

Tabel 11. Nilai SAC bebas *plaintext* dari AES *perfect* SAC *S-box*

Round	**Nilai SAC bebas *plaintext***			
	minimal	maksimal	rata-rata	Error rata-rata
1	0	0,5701	0,1250	0,3750
2	0,48685	0,5185	0,5019	0,0019
3	0,48735	0,51505	0,5000	0,0000
4	0,4851	0,5126	0,5000	0,0000
5	0,4844	0,51305	0,5000	0,0000
6	0,4857	0,51295	0,5000	0,0000
7	0,487	0,51305	0,5000	0,0000
8	0,48505	0,51425	0,5000	0,0000
9	0,4867	0,51325	0,5000	0,0000
10	0,4871	0,5142	0,5000	0,0000

Tabel 12. Nilai SAC bebas *plaintext* dari AES

Round	**Nilai SAC bebas *plaintext***			
	minimal	maksimal	rata-rata	Error rata-rata
1	0	0,57075	0,1266	0,3734
2	0,48685	0,51665	0,5020	0,0020
3	0,48735	0,514	0,5000	0,0000
4	0,4851	0,5137	0,5000	0,0000
5	0,4844	0,51355	0,5000	0,0000
6	0,4857	0,51295	0,5000	0,0000
7	0,487	0,5128	0,5000	0,0000
8	0,48505	0,5131	0,5000	0,0000
9	0,4867	0,51445	0,5000	0,0000
10	0,4871	0,5131	0,5000	0,0000

4. DISKUSI

Hasil penelitian dan pengujian yang telah dilakukan pada *Perfect* SAC *S-box* dan algoritma modifikasi kemudian dianalisis dan dibandingkan dengan *S-box* AES dan algoritma AES asli.

4.1. Analisis Pengujian *Perfect* SAC *S-box*

Pada pengujian AC, nilai maksimal uji AC dari *perfect* SAC *S-box* memiliki nilai maksimal uji AC yang sama dengan *S-box* AES, yaitu sebesar 0,51758. Oleh karena itu, nilai errornya pun sama, yaitu 0,01758, sehingga dapat dikatakan bahwa hasil uji AC *perfect* SAC *S-box* sama baiknya dengan *S-box* AES.

Berdasarkan uji SAC yang telah dilakukan, *Perfect* SAC *S-box* memiliki nilai error sebesar 0,0469. Jika dibandingkan dengan *S-box* AES yang memiliki nilai error SAC sebesar 0.0625, *Perfect* SAC *S-box* memiliki nilai error yang lebih kecil dibanding *S-box* AES sehingga dapat dikatakan *Perfect* SAC *S-box* lebih baik dibanding *S-box* AES pada uji SAC.

Berdasarkan hasil uji LAT, maka nilai bias *Perfect* SAC *S-box* yang dihasilkan adalah ±16/256, yaitu sama dengan nilai bias *S-box* AES yaitu ±16/256. Berdasarkan hasil uji LAT, dapat disimpulkan bahwa *Perfect* SAC *S-box* memiliki kriteria *S-box* yang baik secara kriptografis dan memiliki resistensi yang cukup baik terhadap *linear attack*.

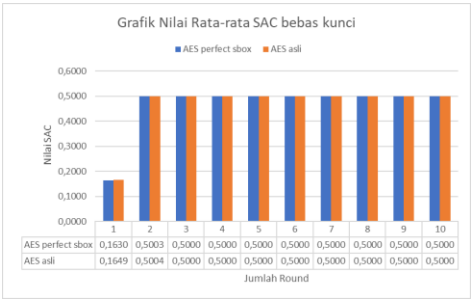
Selanjutnya, untuk hasil uji BIC yang dilakukan terhadap *Perfect* SAC *S-box*, didapatkan hasil nilai BIC maksimal adalah sebesar 0.13498. Jika dibandingkan dengan hasil uji BIC dari *S-box* AES yang memiliki nilai BIC maksimal sebesar 0.13412, maka hasil uji BIC *Perfect* SAC *S-box*, tidak lebih baik dari *S-box* AES.

Untuk pengujian XOR *table*, berdasarkan [21], *S-box* yang memenuhi sifat *Almost Perfect Nonlinear* (APN) adalah *S-box* yang nilai XOR *table* terbesarnya adalah 2, tetapi itu hanya dimiliki oleh *S-box* berukuran ganjil. Sejauh ini belum terdapat *S-box* berukuran genap yang memenuhi sifat APN sehingga *Perfect* SAC *S-box* yang berukuran genap dengan nilai XOR *table* terbesarnya adalah 4 mengindikasikan bahwa *Perfect* SAC *S-box* memiliki resistensi yang cukup baik terhadap *differential attack*.

4.2. Analisis Pengujian SAC Algoritma

Berdasarkan hasil uji SAC dengan variabel bebas kunci, menunjukkan bahwa algoritma asli AES dan algoritma AES dengan *Perfect* SAC *S-box* memiliki hasil uji SAC yang baik. Oleh karena itu pada pengujian SAC dengan variabel bebas kunci, algoritma AES dengan *Perfect* SAC *S-box* juga mempunyai sifat konfusi yang baik. Algoritma AES dengan *Perfect* SAC *S-box* dapat memenuhi uji SAC sejak *round* kedua dengan nilai hasil uji SAC sebesar 0,5003 dan memiliki nilai error 0,0003. Jika dibandingkan hasil uji SAC algoritma asli AES dapat memenuhi uji SAC sejak *round* kedua dengan nilai hasil uji SAC sebesar 0,5004 dan memiliki nilai error 0,0004. Hal ini menunjukkan bahwa hasil uji SAC algoritma AES dengan *Perfect* SAC *S-box* lebik baik dari algoritma asli AES.

Gambar 2. menunjukkan perbandingan hasil uji SAC dengan variabel bebas kunci dari algoritma AES dengan *Perfect* SAC *S-box* dan algoritma asli AES. Pada gambar tersebut memperlihatkan bahwa hasil uji tersebut memiliki nilai yang hampir sama, namun algoritma AES dengan *Perfect* SAC *S-box* memiliki hasil uji SAC yang lebih baik dari algoritma asli AES.

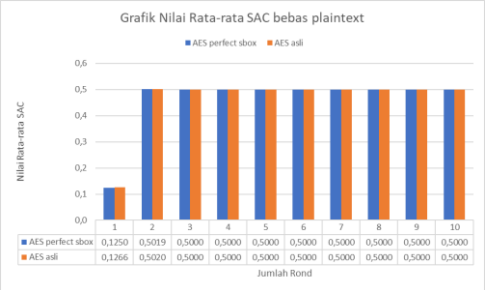


	1	2	3	4	5	6	7	8	9	10
AES perfect sbox	0,1630	0,5003	0,5000	0,5000	0,5000	0,5000	0,5000	0,5000	0,5000	0,5000
AES asli	0,1649	0,5004	0,5000	0,5000	0,5000	0,5000	0,5000	0,5000	0,5000	0,5000

Gambar 2. Grafik Nilai Rata-rata SAC Bebas Kunci

Berdasarkan hasil uji SAC dengan variabel bebas plainteks, menunjukkan bahwa algoritma asli AES dan algoritma AES dengan *Perfect* SAC *S-box* memiliki hasil uji SAC yang baik. Oleh karena itu pada pengujian SAC dengan variabel bebas plainteks, algoritma AES dengan *Perfect* SAC *S-box* juga mempunyai sifat difusi yang baik. Algoritma AES dengan *Perfect* SAC *S-box* dapat memenuhi uji SAC sejak *round* kedua dengan nilai hasil uji SAC sebesar 0,5019 dan memiliki nilai error 0,0019. Jika dibandingkan hasil uji SAC algoritma asli AES dapat memenuhi uji SAC sejak *round* kedua dengan nilai hasil uji SAC sebesar 0,5020 dan memiliki nilai error 0,0020. Hal ini menunjukkan bahwa hasil uji SAC algoritma AES dengan *Perfect* SAC *S-box* lebik baik dari algoritma asli AES.

Gambar 3. menunjukkan perbandingan hasil uji SAC dengan variabel bebas plainteks dari algoritma AES dengan *Perfect* SAC *S-box* dan algoritma asli AES. Pada gambar tersebut memperlihatkan bahwa hasil uji tersebut memiliki nilai yang hampir sama, namun algoritma AES dengan *Perfect* SAC *S-box* memiliki hasil uji SAC yang lebih baik dari algoritma AES asli.



	1	2	3	4	5	6	7	8	9	10
AES perfect sbox	0,1250	0,5019	0,5000	0,5000	0,5000	0,5000	0,5000	0,5000	0,5000	0,5000
AES asli	0,1266	0,5020	0,5000	0,5000	0,5000	0,5000	0,5000	0,5000	0,5000	0,5000

Gambar 3. Grafik Nilai Rata-rata SAC Bebas *Plaintext*

5. KESIMPULAN

Kesimpulan yang dapat diambil berdasarkan hasil penelitian, hasil uji, dan analisis data adalah sebagai berikut bahwa *Perfect* SAC *S-box* yang diimplementasikan pada algoritma AES merupakan *S-box* yang cukup baik secara kriptografis dan terbukti memiliki hasil uji AC, SAC, BIC, XOR *table distribution*, dan LAT *distribution* yang baik dan hampir sama dengan *S-box* AES. Pengujian SAC algoritma modifikasi AES dengan *Perfect* SAC *S-box* yang telah dilakukan menggunakan variabel bebas kunci dan plainteks, algoritma modifikasi AES tersebut dapat memenuhi uji SAC sejak *round* kedua. Hasil uji SAC tersebut memiliki nilai yang hampir sama, namun algoritma AES dengan *Perfect* SAC *S-box* memiliki hasil uji SAC yang lebih baik dari algoritma AES asli. Algoritma modifikasi AES dengan *Perfect* SAC *S-box* memiliki sifat konfusi dan difusi yang lebih baik dari algoritma AES berdasarkan hasil uji SAC algoritma tersebut. Algoritma modifikasi AES dengan *Perfect* SAC *S-box* dapat dijadikan alternatif algoritma enkripsi yang dapat digunakan dalam penerapan kriptografi.

DAFTAR PUSTAKA

[1] Menezes et al, Handbook of Applied Cryptography, Boca Raton: CRC Press LLC, 1997.

[2] NIST, "Advanced Encryption Standard (AES)," Federal Information Processing Standards Publication 197, U.S, 2001.

[3] W. Stallings, Cryptography and Network Security: Principles and Practice Sixth Edition, New Jersey: pearson, 2014.

[4] U. Cavusoglu, "A new approach to design S-box generation algorithm based on genetic algorithm," International Journal of Bio-Inspired Computation, vol. 17 No.1, pp. 52 - 62, 2021.

[5] A. Y. Al-Dweik, I. Hussain, M. S. Saleh dan M. T. Mustafa, "A Novel Method to Generate Key-Dependent S-Boxes with Identical Algebraic Properties," ArXiv abs/1908.09168 (2019), 2019.

[6] Mussher Ahmad et al, "Bijective S-Boxes Method Using Improved Chaotic Map-Based Heuristic Search and Algebraic Group Structures," IEEE Access, Vol. 8, 2020, pp. 110397 - 110411, 2020.

[7] Alamsyah, "A Novel Construction of Perfect Strict Avalanche Criterion S-box using Simple Irreducible Polynomials," Scientific Journal of Informatics, Vol. 7, No. 1, May 2020, 2020.

[8] A. A. T. a. M. M. S. Thwin, "Modification of AES Algorithm by Using Second Key and Modified SubBytes Operation for Text Encryption," dalam International Conference on Computational Science and Technology, Kota Kinabalu, Malaysia, 2018.

[9] Y. S. Chauhan dan T. Sasamal, "Enhancing Security of AES Using Key Dependent Dynamic Sbox," dalam Proceedings of the Fourth International Conference on Communication and Electronics Systems (ICCES 2019), Coimbatore, India, 2019.

[10] H. V. Gamido, A. M. Sison dan R. P. Medina, "Modified AES for Text and Image Encryption," Indonesian Journal of Electrical Engineering and Computer Science, Vol. 11, No. 3, p. pp. 942~948, 2018.

[11] T. S. a. H. M. I. Hussain, "A Projective General Linear Group Based Algorithm for The Construction of Substitution Box for Block Ciphers," Neural Comput and Applic (Springer-Verlag London), vol. 2, pp. 1085-1093, 2013.

[12] M. D. a. I. V. Yucel, "Avalanche and Bit Independence Properties for the Ensembles of Randomly Chosen n x n S-boxes," Turk J Elec Engin, VOL.9, NO.2 2001.

[13] A. A. V. J. a. S. M. M. A. Khan, "A Chaos-Based Substitution Box (S-Box) Design with Improved Differential Approximation Probability (DP)," Iran J Sci Technol Trans Electr Eng, vol. vol. 42, p. p. 219–238, 2018.

[14] Amas, "Pengaruh Penambahan Fungsi Linear dan Fungsi Nonlinear Terhadap kekuatan S-box S1 Clefia," Jurnal Teknologi Informasi, vol. 4, 2020.

[15] D. Zhu, X. Tong, M. Zhang dan Z. Wang, "A New S-Box Generation Method and Advanced Design Based on Combined Chaotic System," Computer and Engineering Science and Symmetry/Asymmetry, vol. 12, 2020.

[16] BSI, "Cryptographic Mechanisms: Recommendations and Key Lengths," 2022.

[17] E.-C. Recommendations, "Algorithms , Key Size and," 2018.

[18] E. Barker dan Q. Dang, "Nist special publication 800-57 part 1, revision 5: Recommendation for key management: Part 1--general," May 2020.

[19] M. D. Selçuk Kavut, "On Some Cryptographic Properties of Rijndael," dalam Information Assurance in Computer Networks: Methods, Models, and Architectures for Network Security, St. Petersburg, Russia, 2001.

[20] S.Aruna dan G.Usha, "Generation of Numerous S-box for Advanced Encryption Standard," International Journal of Recent Technology and Engineering (IJRTE), vol. 8,

no. 2S4, 2019.

[21] D. Davidova, "On properties of bent and almost perfect nonlinear functions," Thesis for the degree of Philosophiae Doctor (PhD), University of Bergen, Norway, 2021.