Jurnal Sistim Informasi dan Teknologi
https://jsisfotek.org/index.php
2021 Vol. 3 No. 2 Hal: 56-63 e-ISSN: 2686-3154

Meningkatkan Keamanan Web Menggunakan Algoritma Advanced Encryption Standard (AES) terhadap Seragan Cross Site Scripting

Yendi Putra[1]✉, Yuhandri Yunus[2], Sumijan[3]
[1,2,3]Universitas Putra Indonesia YPTK Padang
yendiputraraorao@gmail.com

Abstract

In the millennial era, the internet has become a very basic need to support community activities in various fields, one of which is education. SMK Maritim Nusantara in supporting the learning process uses a web-based application called e-learning which is used by teachers and students. The school website has several documents in digital form that must be kept confidential, such as student data, teacher data, student grades. After scanning using the Acunetix WVS 10.5 application, information was obtained about the security holes found on the website https://www.e-learning.smkmn.sch.id, with the results of which there were 8 (eight) attacks with details, 2 (two).) a hight category with the name Cross site scripting (XSS) attack, 4 (four) medium categories with the name HTML form attack without CSRF protection and 2 (two) low categories with the name Password type input attack with auto-complete enabled. The most dangerous attack category / hight is XSS. XSS attack is an attack that inserts malicious code in the form of javascript through an input form that aims to steal cookies and then uses the cookie to enter the web legally so that data can be manipulated and even deleted. For this reason, a strong system is needed to maintain security, confidentiality of school data, one way that can be used is by implementing the Standard Advance Encryption Algorithm (AES), this algorithm has a high level of security and uses little memory in its operation so that it does not burdensome to process and easy to implement. The results of research conducted by applying the AES Algorithm explain that previously there were 2 (two) high category vulnerabilities called XSS attacks, after the implementation of the AES Algorithm, the XSS attack vulnerability was no longer found. Based on the results obtained in the study, it can be concluded that the implementation of the AES Algorithm in tokens can improve the security of the https://www.e-learning.smkmn.sch.id website from XSS attacks.

Keywords: AES Algorithm, XSS, Token, Security, E-learning.

Abstrak

Pada era milenial internet menjadi sebuah kebutuhan yang sangat mendasar guna menunjang aktifitas masyarakat dalam berbagai bidang salah satunya bidang pendidikan. SMK Maritim Nusantara dalam menunjang proses pembelajaran mengunakan aplikasi berbasis web dengan nama e-learning yang digunakan oleh guru dan siswa. Website sekolah memiliki beberapa dokumen dalam bentuk digital yang harus di jaga kerahasianya seperti data siswa, data guru, nilai siswa. Hasil scaning dari aplikasi Acunetix WVS 10.5 diperoleh informasi tentang celah keamanan yang terdapat pada website https://www.e-learning.smkmn.sch.id, dengan hasil diantaranya terdapat 8 (delapan) serangan dengan rincian, 2 (dua) kategori hight dengan nama serangan Cross site scripting (XSS), 4 (empat) kategori medium dengan nama serangan HTML form without CSRF protection dan 2 (dua) kategori low dengan nama serangan Password type input with auto-complete enabled. Kategori serangan yang paling berbahaya/hight yaitu XSS. Serangan XSS adalah serangan yang menyisipkan kode-kode jahat berbentuk javascript melalui form input yang bertujuan untuk mencuri cookie lalu menggunakan cookie tesebut untuk masuk kedalam web secara sah sehingga data dapat dimanipulasi bahkan dihapus. Untuk itu dibutuhan sebuah sistem yang kuat untuk menjaga keamanan, kerahasiaan data-data sekolah, salah satu cara yang dapat digunakan adalah dengan penerapan Algoritma Advance Enkripiton Standart (AES), algoritma ini memiliki tingkat keamanan yang tinggi serta menggunakan memori yang sedikit dalam pengoperasianya sehingga tidak membebani proses dan mudah untuk diterapkan. Hasil penelitian yang dilakukan dengan menerapkan Algoritma AES mejelaskan bahwa sebelumnya terdapat 2 (dua) kerentanan kategori hight dengan nama serangan XSS, setelah implementasi Algoritma AES maka kerentanan serangan XSS tersebut tidak ditemukan lagi. Berdasarkan hasil yang diperoleh dalam penelitian dapat disimpulkan bahwa Implementasi Algoritma AES pada token dapat meningkatkan keamanan website https://www.e-learning.smkmn.sch.id dari serangan XSS.

Kata Kunci : Algoritma AES, XSS, Token, Keamanan, E-learning.

1. Pendahuluan

Cybercrime adalah tindak kejahatan yang dilakukan dengan memanfaatkan teknologi komputer sebagai alat kejahatan utama yang memanfaatakan perkembangan teknologi komputer khususnya internet. Menurut laporan dari Acunetix Web Application Vulnerability tahun 2019, serangan yang terjadi di dunia pada tahun 2019 dari total 100% serangan terdapat 25% serangan dengan menggunakan metode *Cross Site Scripting* (XSS) dengan rincian sebagai berikut: serangan pada Angular JS Template Injection sebesar 0.52%, ikuti dengan serangan DOM XSS sebesar 1,23% dan sisanya XSS 24.5%. Dengan melihat jumlah serangan tersebut maka website-website cukup rentan terhadap serangan. Sedangkan di Indonesia menurut Badan Sandi Siber Negara terdapat 88 juta serangan dalam

Diterima: 16-10-2020 | Revisi: 31-10-2020 | Diterbitkan: 30-06-2021 | DOI: 10.37034/jsisfotek.v3i2.44

rentang waktu 4 bulan terakhir dari januari sampai april 2020, salah satu serangan menargetkan para penguna web aplikasi.

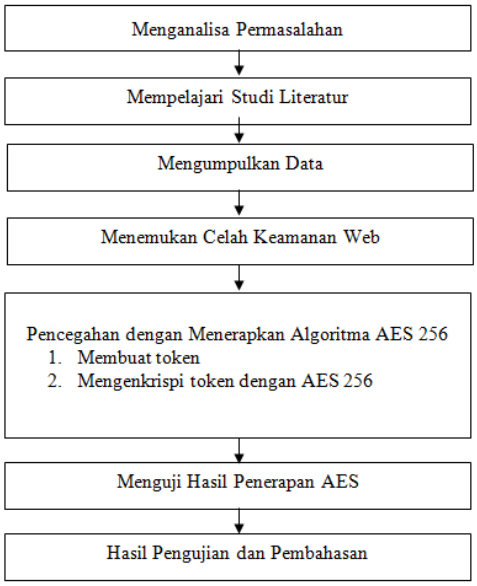
Pada umumnya sekolah-sekolah menggunakan website sebagai media informasi dan media pendukung proses belajar mengajar. Tetapi website yang ada di sekolah-sekolah belum memiliki tingkat kemanan atau proteksi yang kuat, sehingga menimbulkan peluang kejahatan pencurian dan manipulasi data sekolah, siswa serta guru untuk mendapatkan keuntungan oleh pelaku. SMK Maritim Nusantara merupakan sekolah menengah kejurusan bidang kemaritiman yang berada di pesisir pantai Kabupaten Padang Pariaman, dalam mendukung proses pembelajaran SMK maritim nusantara menggunakan aplikasi berbasis web yaitu https://e-learning.smkmn.sch.id/, yang dapat diakses oleh semua siswa dan guru. Pada website sekolah terdapat data-data penting sekolah seperti data siswa, guru, pembelajaran dan data nilai siswa. Setelah penulis melakukan observasi hasilnya menunjukan bahwa website sekolah ini memiliki celah kerentanan yang dapat merugikan sekolah seperti pencurian atau modifikasi data siswa, guru, pembelajaran dan data nilai siswa yang dapat disalah gunakan oleh peretas. Jenis serangan yang ditemuka berupa XSS.

XSS ialah serangan dengan menyuntikan atau menyisipkan code-code jahat berbentuk javascript melalui form input yang bertujuan untuk mencuri cookie lalu menggunakan cookie tesebut untuk masuk kedalam web secara sah [1]. *XSS* memiliki 3 kategori diantaranya, *DOM-Based XSS* cara kerja dengan memanfaatkan javascript untuk memanipulasi objek model, yang berikutnya *Stored On Presistent XSS* bekerja dengan menyintikan javascript ke server dan disimpan permanen dalam database dan yang terakhir mengunakan teknik memantulkan kode berbahaya ke browser yang digunakan korban, cara ini disebut dengan *Refected Non-Persisent XSS* [2], [3]. Apabila hal ini dibiarkan maka akan mendapatkan masalah besar karena bocornya data sekolah ke pihak yang tidak bertanggung jawab. Untuk mengatasi masalah ini dapat cegah dengan menambahkan function php yaitu strip_tag, cara kerjanya dengan menghapus tag html yang menempel pada form inputan tersebut, sehingga *script XSS* tidak di proses melainkan di tampilkan dalam bentuk teks biasa [4]. *Functian strip_tag* juga digunakan pecegahan dapat juga dilakukan dengan pembatasan pada *alphanumeric* Features berupa pembatasan pada Readability, Objects, Events, Methods, Tags, Attributes, Reserved, Functions, Protocol, Letters, Numbers [5]. Pencegahan dari serangan *XSS* juga dapat digunakan dengan menggunakan *metacharacter* khusus yang digunakan dalam sebuah script, selain itu fungsi *html_specialchars()* dapat digunakan untuk merubah *format HTML* diantaranya: 1) & (*ampersand*) menjadi &*amp*, 2) *single quote* menjadi &#039, 3) (*double quote*) menjadi & *quo* [6]. Namun cara diatas masih kurang efektif, langkah yang lebih efektif yaitu menggunakan autentikasi token [7].

Token berfungsi untuk mengakses halaman tertentu, pengguna akan mengirim balik token tersebut sebagai bukti bahwa pengguna sudah berhasi login, apabila tidak cocok maka akan keluar perintah dan apabila cocok maka data bisa diproses [8]. Agar lebih kuat token tersebut dilakukan enkripsi dengan menggunakan *Algoritma* dari Rijndael bernama *Advanced Encryption Standard*. *Algoritma AES* mempunyai 3 jenis panjang kunci diantaranya 127, 192 dan 256, serta mempunyai panjang blok 128 bit [9]. *Algoritma AES* berfungsi untuk *enkripsi* dan *deskripsi* informasi atau data, dengan menggunakan roude atau kegiatan yang, selain itu *AES* lebih mudah diterapkan, pengunaan memori relative rendan serta efisien dari segi [10]. Teknik menggunakan gabungan dua metode antara *Steganografi LSB* dengan *Algoritma AES*. Hasil penelitian ini menitik beratkan pada aspek *imperceptibility* dan *recovery* terhadap metode modifikasi LSB yang bertujuan untuk meningkatkan keamanan dari data [11]. Sedangkan penelitian yang dilakukan dalam meningkatkan keamanan apalikasi e-commrce dengan menerpakan *Algoritma* AES pada data costomer [12]. Hasil penelitan menjelaskan *Algoritma* AES pada database lebih aman karena data rahasia dienkripsi dengan mengukan *key* 128 dan 10 *round* sehingga sulit untuk dipecahkan serta memiliki kecepatan tinggi karena menggunakan memori yang kecil. Maka, penelitian ini bertujuan mengukur tingkat keamanan website e-learning SMK Maritim Nusantara dari serangan *XSS* selanjutnya implementasi *Algoritma AES* pada token untuk meningkatkan keamanan website.

2. Metodologi Penelitian

Langkah-langkah yang diambil penulis dalam penelitian ini tidak melenceng dari pokok pembahasan dan lebih mudah dipahami, maka susunan langkah-langkah dibuat secara sistematis sehingga mudah dijadikan pedoman. Urutan langkah-langkah yang akan dikerjakan pada penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Kerangka Kerja Penelitian

Uraian kerangka kerja adalah penjabaran secara rinci tentang masing-masing kerang kakerja yang telah disusun agar penelitian yang dilakukan terstruktur. Berdasarkan Gambar 1 dari kerangka kerja diatas, maka masing-masing tahapan dapat dijelaskan sebagai berikut:

a. Menganalisa Permasalahan bertujuan agar penelitian yang akan dilakukan dapat memahami masalah yang telah ditentukan batasan masalahnya.

b. Mempelajari Literatur dilakukan agar mempunyai landasan baik secara teoritis yang benar dan jelas yang telah dijelaskan oleh para peneliti dan ahli sebelumnya.

c. Mengumpulkan data dilakukan untuk dapat memperoleh informasi data-data yang dibutuhkan dalam penelitian rangka mencapai tujuan penelitian.

d. Menemukan celah keamanan Web dengan melakukan scaning mengunakan aplikasi acunetix, pada penelitian ini peneliti hanya melakukan pengaman website dari serangan *XSS.*

e. Pencegahan dengan menerapkan *Algoritma AES*, Langkah-langkah yang dilakukan dalam adalah sebagai berikut:

- Membuat *token* dengan menggunakan fungsi dari *Base64Encode* dan gabungan antara header, playload dan signature [13];
- Mengenkrispi *token* dengan *Algoritma AES* agar serangan dari *XSS* bisa dihindari.

f. Menguji Hasil Penerapan *Algoritma AES* pada website *e-learning* SMK Maritim Nusantara dengan menggunakan Aplikasi *Acunetix*.

g. Hasil Pengujian dan Pembahasan menjelaskan tentang hasil dari penerapan dan pengujian *Algorimtma AES* pada website *e-learning* SMK Maritim Nusantara. Hasil dari penerapan *Algorima AES* tersebut akan dibandingkan dengan data-data sebelum penerapan *Algoritma AES*.

2.1. Data

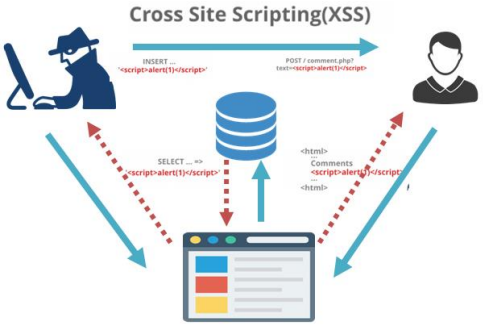
Pada aplikasi e-learning menggunakan server yang disewa pertahun pada penyedia layanan hosting dewaweb.com. Berikut ini data rinci server dan hosting e-learning SMK Maritim Nusantara dapat dilihat pada Tabel 1

Tabel 1. Data Rinci Server Dan Hosting E-Learing SMK Maritim Nusantara

No	Nama	Keterangan
1.	Penyedia Layanan Hosting	Dewaweb.com
2.	Nama paket layanan	Cloud hosting warior
3.	Teknologi Server	LiteSpeed
4.	Storage	3 GB
5.	Kecepatan CPU	1 Core
6.	Memory	512 Mb
7.	Addon domain	6
8.	IP	104.24.110.244
9.	Tanggal aktif	14 Maret 2019
10	Tanggal nonaktif	13 Maret 2021
11.	Database	Mysql
12.	Bahasa Pemrograman	PHP
13.	Domain ID	PANDI-DO1298761
14.	CDN	Cloudflare

2.2. *Cross Site Scripting*

Cross site scripting disingkat (*XSS*) adalah jenis kerentanan yang terjadi pada sebuah web dinamis karena web yang dibangun tidak bisa memfilter masukan yang dikirim oleh penyerang, sehingga penyerang dengan mudah menginjeksi atau menyuntikan code jahat berupa *javascript* pada halaman web dengan tujuan untuk mendapatkan *cookie* dan *session* untuk digunakan sebagai hak akses yang sah kedalam situs tersebut[3]. Cara Kerja Serangan *XSS* Gambar 2.



Gambar 2. Cara Kerja Serangan XSS

2.3. Json Web Token (JWT)

JWT memiliki tiga bagian diantaranya *header*, *payload*, serta *signature*, antara bagian satu dan lainya dihubungkan dengan titik (".") diantaranya [14]:

a. *Header*

Header terdiri memiliki dua komponen, yaitu Algoritma hashing seperti *HMAC SHA-256* serta tipe *token*.

b. *Payload*

Payload ialah bertugas untuk menjelaskan tentang metadata atau suatu entitas. Terdapat tiga jenis klaim, yaitu *private, public, dan reserved*.

c. *Signature*

Signature berisi hash terdiri dari komponen-komponen header, payload serta kunci rahasia.

Token yang dihasilkan dari *JWT* dibangun dengan menggunakan rumus atau formula sebagai berikut [15]:

$$\text{Token} = \text{f(Base64Encode)} \sum_{n=\alpha,\beta}^{\infty} (\text{header}, \text{playoad}, \text{signature})$$

Dimana:

f(Base64Encode) = Fungsi untuk melakukan penyandian dengan Base64Encode

$\sum_{n=\alpha,\beta}^{\infty}$ = Perulangan penjumlahan bilangan

(header, playoad, signature) = Bagian kepala, bagian data, bagian kunci

2.4. *Advanced Encription Standard* (*AES*)

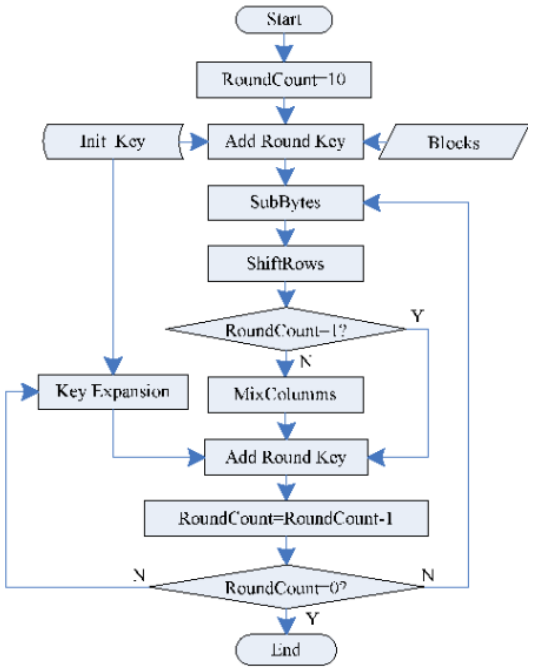
Kriptografi terdiri dari dua suku kata yaitu (*kryptos*) yang bermakna rahasia dan (*graphein*) yang bermakna tulisan, jadi kriptografi memiliki makna tulisan rahasia (Mollin, 2007).

Kriptografi dikelompokan menjadi dua diantaranya berdasarkan kunci yang digunakan, yaitu *algoritma simetri* (mengunakan satu kunci untuk proses *enkripsi* dan *dekripsi*), sedangkan *Algoritma asimetri* (memiliki kunci yang berbeda saat proses *enkripsi* dan *dekripsi* serta fungsi *hash* (Ariyus, 2008) [16]. Sedangkan karakteristik kriptografi digolongkan menjadi dua bagian yaitu berdasarkan tipe operasi pada *enkripsi* dan dekripsi (teknik *substitusi* dan teknik *permutasi*) selain itu proses pengolahan pesan berupa *block chipher* dan *stram chipher*.

Pada tahun 1998 diadakan sayembara NIST untuk melahirkan kriptografi baru untuk menganti *Algoritma DES*. Standard tersebut nantinya akan diberi nama *Advanced Encription Standart* (AES). Adapun syarat untuk sebuah AES ialah seluruh rancangan bersifat public. NIST memilih 5 finalis yang memenuhi syarat dari 15 proposal yang masuk, setelah itu dilakukan voting dan hasilnya Rijndael Vincent Rijmen dan Joan Daemen Belgia, dinobatkan sebagai pemenang dengan 86 suara. Pada tahun 2001 tepatnya bulan november, NIST mengumumkan bahwasanya Rijndael (dibaca: Rhine-doll) ditetapkan sebagai standard kriptografi sah dan dapat digunakan. Laila Mustika, (2019) Algoritma AES menggunakan substitusi dan permutasi serta sejumlah putaran, setiap putaran memiliki kunci internal yang berbeda disebut dengan round key dan beroperasi dalam orientasi byte.

Algoritma Advance Enkription Standart

a. *AddRoundKey*
b. *Round*
   - *SubBytes*
   - *ShiftRows*
   - *MixColumns*.
   - *AddRoundKey*
c. Final round
   - *SubBytes*
   - *ShiftRows*
   - *AddRoundKey*



Gambar 3. Flowchart Algoritma AES

Program AES

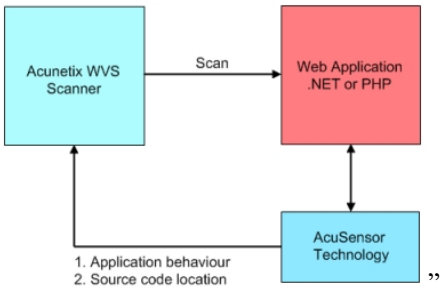
Input: P,K
Output: CP
Initialization
  Nr.w ←EkspansiKunci(K)
  CP ← P
  RoundCount ← 10
  AddRoundKey(CP, round_key[0])
    for i = 1→Nr do
        SubBytes(CP)
        ShiftRows(CP)
        MixColumns(CP)
      AddRoundKey(CP, round_key[i])
    end for
end

Gambar 4. Algoritma AES

3. Hasil dan Pembahasan

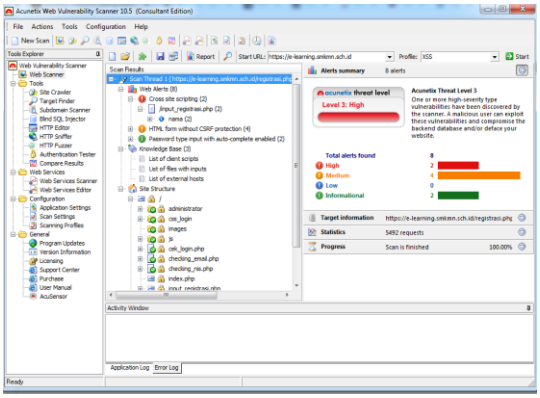
Analisa kerentanan dalam penelitian ini dilakukan dengan melakukan scaning pada website e-learning SMK Maritim Nusantara dengan menggunakan aplikasi

Acunetix Web Vulnerability Scanner 10.5. Cara kerja aplikasi Acunetix disajikan pada Gambar 2.



Gambar 5. Cara Kerja Aplikasi Acunetix

Hasil dari proses scaning website e-learning SMK Maritim Nusantara menggunakan aplikasi Acunetix memberikan beberpa informasi diantaranya terlihat pada Gambar 3.



Gambar 6. Hasil Scaning web e-learning SMK Maritim mengunakan Aplikasi Acunetix

Tabel 2. Data Rincian Hasil Scaning Web e-learning Menggunakan Aplikasi Acunetix

No	Indikasi File	Kerentanan	Kategori Resiko
1.	/input_registrasi.php. nama	*Cross site scripting*	*Hight*
2.	/input_registrasi.php nama	*Cross site scripting*	*Hight*
3.	/administrator.	*HTML form without CSRF protection*	*Medium*
4.	/login.php	*HTML form without CSRF protection*	*Medium*
5.	/registrasi.php	*HTML form without CSRF protection*	*Medium*
6.	/login.php	*HTML form without CSRF protection*	*Medium*
7.	/administrator.php	*Password type input with auto-complete enabled*	*Low*
8.	/login.php	*Password type input with auto-complete enabled*	*Low*

Hasil scaning menunjukan bahwa terdapat 2 (dua) kerentanan dengan level *hight* yaitu serangan *Cross Site Scrip*t, serangan ini sangat berbahaya sehingga harus di antisipasi, cara mengatasinya dengan mengunakan token untuk otentikasi, agar token lebih kuat maka dilakukan engkripsi mengunakan algoritma AES

3.1. Pembuatan Token

Token terdiri dari tiga bagian diantarnya. *header*, *playload* dan *signature*, ketiga komponen ini di *encode* mengunakan *base64encode* lalu digabungkan menjadi satu dengan tanda (.) titik sebagai pemisah. Cara pembautan token sebagai berikut

a. Pembuatan *Header*

```
{
  "typ": "JWT",
  "alg": "HS256"
}
```

Hasil dari *encode* mengunakan *base64encode* pada *header* adalah "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9"

b. Pembuatan *Playload*

```
{
 "sub": "1234567890",
 "name": "John Doe",
 "iat": 1516239022
}
```

Hasil dari *encode* mengunakan *base64encode* pada *playload* adalah eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ"

c. *Signature* dihasilkan dari pengabunga *header* dan *playload* dengan penambahan *secret key* untuk mengamankan *token*, *secret key* yang dimasukan adalah "sanditokenaes128". Hasil dari *encode* adalah "goeCWV2u8ee5-gqj-L0B6Y9YgIZcfMKY2Ad2Lv4V-c"

JWT dihasilkan dengan mengabungkan header, playload dan signature dimana secretkey yang digunakan adalah ="sanditokenaes128" . hasilnya sebagai berikut."eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.goeCWV2u8ee5-gqj-2L0B6Y9YgIZcfMKY2Ad2Lv4V-c"

3.2. Implementasi Algoritma AES Pada Secret Key Token

Token yang telah dibuat mengunakan JWT agar lebih kuat dilakukan enkripsi mengunakan *Algoritma AES* dimana *secretkey* menjadi *plaintext* pada *Algoritma AES*. Adapun langkah-langkah sebagai berikut.

Terlebih dahulu ditetapkan *secretkey* sebagai *plaintext*, lalu ditetapkan

Plaintext : sanditokenaes128
*Cipher*key : kuncirahasiantim

Konversikan *plaintext* dan *cipherkey* text dari ASCII ke *hexadecimal* lalu disimpan pada array of bytes dua dimensi yang berukuran 4 x 4. Data masukan, in0, in1, …, in15 disalin ke dalam array state (direalisasikan

oleh fungsi CopyPlaintext ToState(state, plaintext)) seperti Tabel 3.

Tabel 3. Konvesi Plaintext dan Cipherkey dari ASCII ke Hexadecimal

Plaintext			
73	69	65	73
61	74	6e	31
6e	6f	61	32
64	6b	65	38

Cipherkey			
6b	72	73	74
75	61	69	74
6e	68	61	69
69	61	6e	6d

a. *AddRoundKey* yaitu melakukan *XOR* antara *plainteks* dengan *cipherkey*

Tabel 4. XOR plainteks dengan ciperkey

Plaintext			
73	69	65	73
61	74	6e	31
6e	6f	61	32
64	6b	65	38

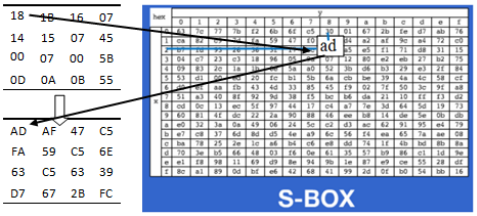
Cipherkey			
6b	72	73	74
75	61	69	74
6e	68	61	69
69	61	6e	6d

Key Adition			
18	1B	16	07
14	15	07	45
00	07	00	5B
0D	0A	0B	55

b. *Round* yaitu melakukan putaran dimana Nr – 1 kali. Proses yang dikerjakan pada setiap putaran adalah

a. *SubBytes* yaitu melakukan substitusi mengunakan tabel (S-box).

Tabel 5. SubBytes mengunakan Tabel S-BOX



b. *ShiftRows* yaitu pergeseran secara wrapping (siklik) pada 3 baris terakhir dari array state, jumlah geseran tergantung nilai baris(r) dimana r=1 digeser 1 byte kekiri, r=2 digeser 2 byte kekiri, r=3 digeser 3 byte kekiri sedangkan r=0 tidak digeser, hasil pergeseran seperti tabel 5

Tabel 6. ShiftRows Array State

AD	AF	47	C5
FA	59	C5	6E
63	C5	63	39
D7	67	2B	FC

⇒

AD	AF	47	C5
59	C5	6E	FA
63	39	63	C5
FC	D7	67	2B

c. *MixColumns* pada array state digunakan sebagai *polynomial* yang berada pada GF(28) dan lalu dilakukan perkalian dengan modulo x4 +1 , dengan suatu *polynomial*

Tabel 7. Hasil Transformasi ShiftRows() Sebelumnya

AD	AF	47	C5
FA	59	C5	6E
63	C5	63	39
D7	67	2B	FC

d. *AddRoundKey* melakukan XOR antara *round key* dengan *array state*

Tabel 8. Hasil *AddRoundkey*

round key			
6b	72	73	74
75	61	69	74
6e	68	61	69
69	61	6e	6d

array state			
74	92	01	93
69	F9	00	F9
6d	3C	00	3C
74	92	00	92

6b		74	F8
75	⊕	69	8C
6e		6d	52
69		74	FB

F8	8A	F9	8D
8C	ED	84	F0
52	3A	5B	32
FB	9A	F4	99

c. *Final round* Pada putaran akhir tidak lagi melakukan *mixclom* melainkan setelah *sihiftrow* langsung *addrounkey* hasil seperti Tabel 7.

Tabel 9. Hasil Enkripsi Mengunakan AES

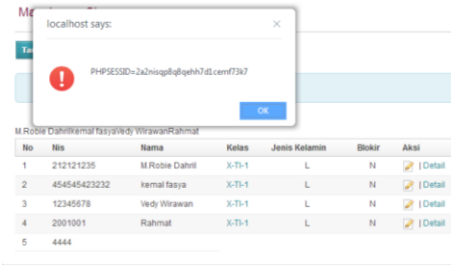
Chipertex			
D5	81	92	C5
E6	A3	21	A2
AF	DF	6C	E3
13	74	95	0F

Chipertext yang dihasilkan melalui *Algoritma AES* di jadikan *secretkey* pada *signature* token, sehingga token yang dihasilkan sebagai berikut “eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOjEyMzQ1Njc4OTAsIm5hbWUiOiJKb2huIERvZSIsImlhdCI6MTUxNjIzOTAyMn0.Dn1t0Q0P_73kiF3qeRXIUUgBMytjuClI3EWiGw4ifoQ”

3.3. Hasil Pengujian

a. Sebelum Implementasi Algoritma AES

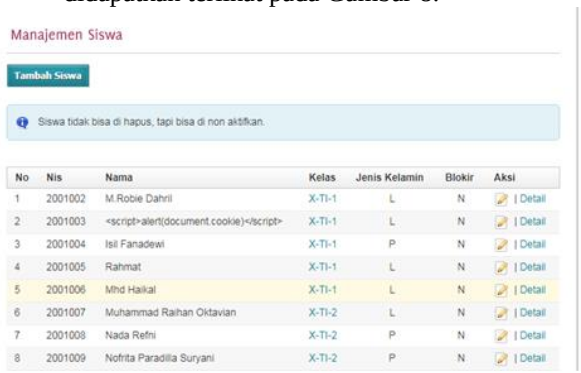
Uji coba dengan memasukan script jahat pada form *<script>alert(document.cookie)</script>* dan hasilnya cookie dengan mudah didapatkan terlihat pada Gambar 7.



Gambar 7. Uji Coba Memasukan *Script* Berbahaya Sebelum *Implementasi Algoritma AES*

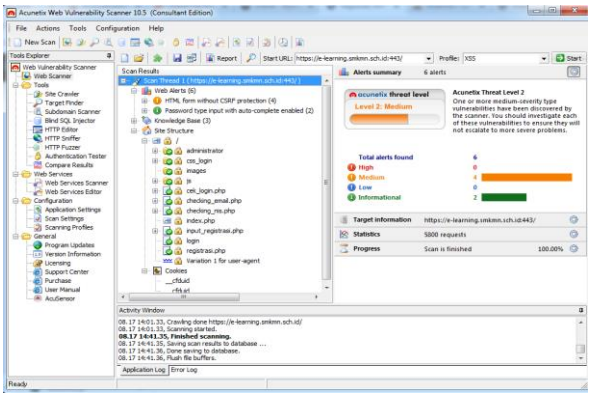
b. Setelah Penerpaan Algoritma AES

Uji coba dengan memasukan script *<script>alert(document.cookie)</script>* dan hasilnya script tersebut tidak bisa dieksekusi melainkan ditampilkan sesuai karakternya didapatkan terlihat pada Gambar 8.



Gambar 8. Uji Coba Memasukan *Script* Berbahaya Setelah *Implementasi Algoritma AES*

penerapan Algoritma AES maka kerentanan yang semula berjumlah 8 (delapan) berkurang menjadi 6 (enam) dan kerentanan kategori hight sudah bisa teratasi, hasil scaning menggunakan Acunetix dapat dilihat pada Gambar 9.



Gambar 9. Hasil Scaning Setelah Implementasi Algoritma AES Mengunakan Aplikasi Acunetix

Selanjutnya dilakukan pengujian menggunakan aplikasi acunetix mendapatkan tingkat kerentanan yang sebelumnya terdapat tingkat kerentanan hight yaitu serangan XSS, dengan

Agar lebih jelas hasil perbandingan antara sebelum dan setelah *Implementasi Algoritma AES* dapat dilihat pada Tabel 10.

Tabel 10. Perbandigan sebelum dan sesusah impelementasi Algoritma AES

	Sebelum Implementasi Algoritma AES			Setelah Implementasi Algoritma AES		
No	Indikasi File	Kerentanan	Kategori Resiko	Indikasi File	Kerentanan	Kategori Resiko
1.	/input_registrasi.php. nama	*Cross site scripting*	*Hight*	Berhasil diatasi	Berhasil diatasi	Berhasil diatasi
2.	/input_registrasi.php nama	*Cross site scripting*	*Hight*	Berhasil diatasi	Berhasil diatasi	Berhasil diatasi
3.	/administrator.	*HTML form without CSRF protection*	*Medium*	/administrator.	*HTML form without CSRF protection*	*Medium*
4.	/login.php	*HTML form without CSRF protection*	*Medium*	/login.php	*HTML form without CSRF protection*	*Medium*
5.	/registrasi.php	*HTML form without CSRF protection*	*Medium*	/registrasi.php	*HTML form without CSRF protection*	*Medium*
6.	/login.php	*HTML form without CSRF protection*	*Medium*	/login.php	*HTML form without CSRF protection*	*Medium*
7.	/administrator.php	*Password type input with auto-complete enabled*	*Low*	/administrator.ph	*Password type input with auto-complete enabled*	*Low*
8.	/login.php	*Password type input with auto-complete enabled*	*Low*	/login.php	*Password type input with auto-complete enabled*	*Low*

4. Kesimpulan

Berdasarkan hasil dari penelitian tentang Implementasi *Algoritma AES* pada *token* dalam meningkatkan keamanan website dari seragan *XSS* yang telah dilaksanakan dengan melakukan pengujian mengunakan Aplikasi Acunetix WVS 10.5 pada website https://www.e-learning.smkmn.sch.id. Memberikan informasi diantaranya terdapat 8 (delapan) serangan dengan rincian, 2 (dua) kategori *hight* dengan nama serangan *XSS*, 4 (empat) kategori *medium* dengan nama serangan *HTML form without CSRF protection* dan 2 (dua) kategori *low* dengan nama serangan *Password type* input *with auto-complete enabled*. Hasil setelah menerapkan *Algoritma AES* mejelaskan bahwa sebelumnya terdapat 2 (dua) kerentanan kategori *hight* dengan nama *Serangan XSS*, setelah implementasi Algoritma AES maka kerentanan dari serangan *XSS* dapat diatasi. *Algoritma AES* mudah untuk diterapkan, memiliki tingkat keamanan yang tinggi serta mengunakan memori yang sedikit dalam pengoperasianya sehingga tidak membebani proses dan ukuran file. Pada pengembangan aplikasi berikutnya agar session disimpan ke dalam database supaya mudah di manajemen.

Daftar Rujukan

[1] Putra, S. S. H. (2017). Penanggulangan Serangan XSS , CSRF , SQL Injection Menggunakan Metode Blackbox Pada Marketplace IVENMU. *Jurnal Pendidikan dan Teknologi Informasi*, *4*(2), 289–300.

[2] Marashdih, A. W., & Zaaba, Z. F. (2017). Cross Site Scripting: Removing Approaches in Web Application. *Procedia Computer*

Science, *124*, 647–655. DOI: http://doi.org/10.1016/j.procs.2017.12.201 .

[3] Marashdih, A. W., Zaaba, Z. F., & Omer, H. K. (2017). Web Security: Detection of Cross Site Scripting in PHP Web Application using Genetic Algorithm. *International Journal of Advanced Computer Science and Applications (IJACSA)*, *8*(5). DOI: http://doi.org/10.14569/ijacsa.2017.080509 .

[4] Mohammadi, M., Chu, B-T., & Lipford, H. R. (2018). Automated Detecting and Repair of Cross-Site Scripting Vulnerabilities. *Cornell Unversity*.

[5] G, K. N., S. Sahana, S., & Santhosh, K. B. J. (2019). Detection and Avoidance of Web Vulnerability Using XSS. *International Journal of Recent Technology and Engineering (IJRTE)*, *8*(2), 1737–1740. DOI: http://doi.org/10.35940/ijrte.B1039.078219 .

[6] Fang, Y., Huang, C., Xu, Y., & Li, Y. (2019). RLXSS: Optimizing XSS Detection Model to Defend Against Adversarial Attacks Based on Reinforcement Learning. *Future Internet*, *11*(8). DOI: http://doi.org/10.3390/fi11080177 .

[7] Yulianingsih, Y. (2017). Melindungi Aplikasi dari Serangan Cross Site Scripting (XSS) dengan Metode Metacharacter. *Journal Nasional Teknologi & Sistem Informasi*, *3*(1), 83–88. DOI: http://doi.org/10.25077/teknosi.v3i1.2017.83-88 .

[8] Rahmatulloh, A., Sulastri, H., & Nugroho, R. (2018). Keamanan RESTful Web Service Menggunakan JSON Web Token (JWT) HMAC SHA-512. *Jurnal Nasional Teknologi Elektro dan Teknologi Informasi*, *7*(2). DOI: http://dx.doi.org/10.22146/jnteti.v7i2.417 .

[9] Aris., Sahara, S., Aini, N., Ajija, M. T., & Mauna, R. N. (2017). Implementasi Kriptografi Algoritma AES Serta Algoritma Kompresi Huffman dengan Menggunakan Pemograman PHP. Koferensi Nasional Sistem & Informatika, 2(1), 225–230.

[10] Prameshwari, A., & Sastra, N. P. (2018). Implementasi Algoritma Advanced Encryption Standard (AES) 128 Untuk Enkripsi dan Dekripsi File Dokumen. *Jurnal Eksplora Informatika*, *8*(1). DOI: http://doi.org/10.30864/eksplora.v8i1.139 .

[11] Anwar, S. (2017). Implementasi Pengamanan Data dan Informasi dengan Metode Steganografi LSB dan Algoritma Kriptografi AES. *Jurnal Format*, *6*(1).

[12] Santoso, K. I., & Priyoatmoko, W. (2016). Pengamanan Data Mysql pada E-Commerce dengan Algoritma AES 256. *Seminar Nasional Sistem Informasi Indonesia*, *1*(1).

[13] Wiguna, B. S., Kusyanti, A., & Yahya, W. (2018). Implementasi Algoritme Blake2s pada JSON Web Token (JWT) sebagai Algoritme Hashing untuk Mekanisme Autentikasi Layanan REST-API. *JPTIIK Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(12), 6269–6276.

[14] Gunawan, R., & Rahmatulloh, A. (2019). JSON Web Token (JWT) untuk Authentication pada Interoperabilitas Arsitektur berbasis RESTful Web Service. *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, 5(1). DOI: http://dx.doi.org/10.26418/jp.v5i1.27232 .

[15] Budianto, W., Amini, S., & Ariyani, P. F. (2017). Aplikasi Pengamanan Dokumen Digital Menggunakan Algoritma Kriptografi Advanced Encryption Standard (AES-128), Kompresi Huffman Dan Steganografi End of File (EoF) Berbasis Desktop Pada Cv. Karya Perdana. *Seminar Nasional Teknologi dan Informatika* (*Prosiding SNATIF)*.

[16] Mustika, L. (2020). Implementasi Algoritma AES Untuk Pengamanan Login dan Data Customer Pada E-Commerce Berbasis Web. *JURIKOM (Jurnal Riset Komputer)*, 7(1). DOI: http://dx.doi.org/10.30865/jurikom.v7i1.1943 .