JURNAL AKADEMIKA

Jurnal Hasil Penelitian

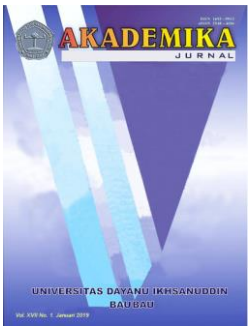
https://www. ejournal.lppmunidayan.ac.id/index.php/akd

e-ISSN : 2548-4184
P-ISSN : 1693-9913

Keywords**:** *URL, Cryptography, AES-128, SQL Injection.*

Kata kunci: URL, Kriptografi, AES-128, Injeksi SQL.

Korespondensi Penulis:
Email: hamidwijaya35@gmail..com



PENERBIT

Lembaga Penelitian dan Pengabdian pada Masyarakat Universitas Dayanu Ikhsanuddin Baubau.

Alamat: Jl. Dayanu Ikhsanuddin No. 124 Baubau

IMPLEMENTASI KRIPTOGRAFI AES-128 UNTUK MENGAMANKAN URL (UNIFORM RESOURCE LOCATOR) DARI SQL INJECTION

Hamid Wijaya[1)]

[1)] Program Studi Teknik Informatika Universitas Dayanu Ikhsanuddin, Baubau, Indonesia.

Dikirim: 11/12/2019;
Direvisi: 15/01/2020;
Disetujui: 30/01/2020.

Abstract

On a website there is a URL (Uniform Resource Locator) that contains the server address, protocol and file path used to provide information to users. At that URL, if no special handling is done in the form of encryption, it will be very vulnerable to hacking. One form of hacking that is done on a URL is to do a SQL injection attack. The form of SQL injection attacks is the act of entering a special code in the website URL so that it can make changes to the contents of the database. For that we need a way to prevent SQL injection attacks by applying cryptography. Cryptography used in this research is by applying AES-128 cryptography. To test the application of AES-128 cryptography the SQLmap application is used. The results of the application of AES-128 cryptography are URLs that have been encrypted so that they are safe from SQL injection attacks.

Intisari

Pada sebuah website terdapat URL (*Uniform Resource Locator*) yang berisikan alamat server, protocol dan path file yang digunakan untuk memberikan informasi kepada pengguna. Pada URL tersebut, jika tidak dilakukan penanganan khusus berupa enkripsi, akan sangat rentan untuk dilakukan peretasan. Salah satu bentuk peretasan yang dilakukan pada URL yaitu dengan melakukan serangan *SQL injection*. Bentuk serangan *SQL injection* berupa tindakan memasukkan kode khusus pada URL website sehingga dapat melakukan perubahan pada isi database. Untuk itu diperlukan suatu cara untuk mencegah serangan *SQL injection* yaitu dengan cara penerapan kriptografi. Kriptografi yang digunakan dalam penelitian ini yaitu dengan menerapkan kriptografi AES-128. Untuk menguji penerapan kriptografi AES-128 digunakan aplikasi *SQLmap*. Hasil dari penerapan kriptografi AES-128 yaitu berupa URL yang telah dienkripsi sehingga aman dari serangan *SQL injection*.

1. PENDAHULUAN

Website merupakan suatu sarana yang digunakan untuk menyebarkan informasi melalui media internet, baik berupa teks, suara, gambar dan video. Seiring dengan penggunaan website yang semakin meluas, berbagai dampak tindak kejahatan terhadap website semakin meluas juga, seperti pencurian data, manipulasi data dari suatu website oleh orang yang tidak bertanggung jawab [1].

Dalam suatu website terdapat dokumen-dokumen atau informasi yang dihubungkan melalui alamat URL (*Uniform Resource Locator*). Pada URL terdapat informasi berupa alamat server, lokasi dan nama dokumen tersebut tersimpan. URL website juga dapat digunakan untuk memberikan berbagai macam perintah terhadap basis data yang terdapat pada server website tersebut. Oleh karena itu, URL website sering digunakan sebagai media untuk melakukan tindakan kejahatan terhadap suatu website. Berbagai macam cara dapat dilakukan untuk dapat meretas suatu website [2]. Salah satu cara yang banyak digunakan adalah dengan menggunakan metode SQL injection.

Metode *SQL injection* digunakan untuk memasukan perintah atau kode khusus SQL sebagai masukkan pada suatu website untuk mendapatkan informasi server maupun akses ke dalam basis data [3]. Salah satu cara yang dapat digunakan untuk mengamankan suatu URL website dari serangan *SQL injection* adalah dengan implementasi kriptografi.

Kriptografi merupakan suatu ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan data dan informasi seperti keabsahan data, integritas data, serta autentifikasi data [4]. Dalam sistem kriptografi terdapat fasilitas untuk mengkonversikan pesan jelas (plainteks) ke pesan yang telah disandikan (cipherteks). Proses konversi ini disebut enkripsi. Sebaliknya, proses menerjemahkan cipherteks menjadi plainteks disebut dengan dekripsi [5].

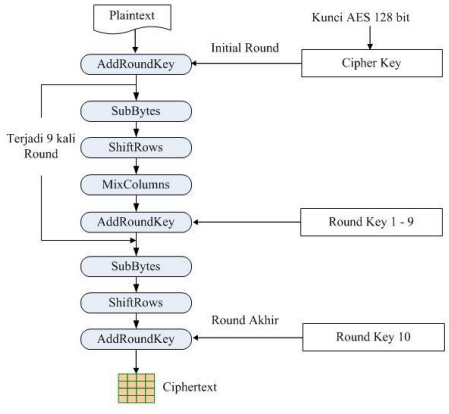
Pada tahun 2001, *National Institute of Standards and Technology* (NIST) sebagai agensi departemen perdagangan Amerika Serikat menetapkan sebuah standard kriptografi yang baru yaitu Algoritma Rijandel dan ditetapkan sebagai *Advanced Encryption Standard* (AES) [6]. AES secara garis besar beroperasi pada blok 128-bit atau 16 karakter sehingga disebut AES-128. Pada URL terdapat barisan karakter yang berukuran kurang atau lebih dari 16 karakter. Akan tetapi, AES memiliki keunggulan yaitu dapat digunakan untuk penyandian dengan melakukan enkripsi perblok (128 bit) secara pararel untuk memudahkan proses enkripsi maupun dekripsi. Berdasarkan keunggulan tersebut, maka dilakukan penelitian tentang implementasi kriptografi AES-128 untuk mengamankan URL (*Uniform Resource Locator*) dari *SQL Injection*.

2. METODE

AES merupakan *block* kode simetris yang menggantikan Algoritma DES (*Data Encryption Standard*). Algoritma AES memiliki ukuran block yang tetap yaitu sepanjang 128 bit dengan panjang kunci yang berbeda-beda [7]. Untuk Kunci AES-128 menggunakan proses perulangan yang disebut sebagai round yaitu sebanyak 10 kali putaran dengan pola matriks 4 x 4 dimana setiap pola matriks terdiri atas 1 byte atau 8 bit untuk melakukan enkripsi dan dekripsi [8].

2.1 Proses Enkripsi

Proses enkripsi algoritma AES-128 terdiri dari 4 jenis transformasi *bytes*, yaitu *SubBytes*, *ShiftRows*, *MixColumns*, dan *AddRoundKey*. Pada Awal proses enkripsi, *state* (dimensi) akan mengalami transformasi *byte AddRoundKey*. Setelah itu, *state* akan mengalami transformasi *SubBytes*, *ShiftRows*, *MixColumns*, dan *AddRoundKey* secara berulang sebanyak Nr (nilai *round*). Proses ini disebut sebagai *round function*. Pada round terakhir, proses yang dilakukan berbeda dari sebelumnya dimana *state* tidak mengalami transformasi *MixColumns* [9]. Proses enkripsi AES dapat dilihat pada gambar 1 berikut ini.



Gambar 1. Proses Enkripsi AES-128

Secara garis besar proses enkripsi AES-128 dengan kunci 128 bit adalah sebagai berikut:

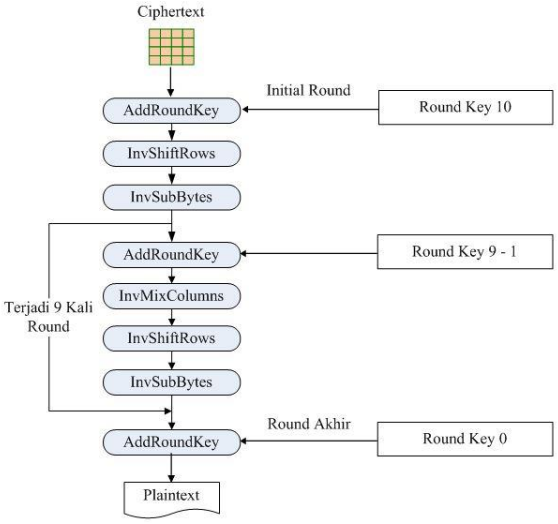
a. *AddRoundKey*: melakukan XOR antara *state* awal (plainteks) dengan *cipher key*. Pada Tahap ini disebut juga *initial round*.
b. *Round* : Putaran sebanyak Nr – 1 kali. Proses yang dilakukan pada setiap putaran adalah:
   1) *SubBytes*: substitusi *byte* dengan menggunakan tabel substitusi (S-box).
   2) *ShiftRows*: pergeseran baris-baris *array state* secara *wrapping*.
c. *MixColumns*: mengacak data pada masing-masing kolom *array state* dengan persamaan sebagai berikut:

$$A(x) = \{03\}x^2 + \{01\}x^2 + \{01\}x^2 + \{02\} \quad (1)$$

d. *AddRoundKey*: melakukan XOR antara *state* sekarang *round key*.
e. *Final Round*: proses untuk putaran terakhir antara lain:
   1) *SubBytes*
   2) *ShiftRows*
   3) *AddRoundKey*
f. Pada proses terakhir akan menghasilkan karakter atau teks yang berbentuk *chipertext*.

2.2 Proses Dekripsi

Untuk proses Dekripsi pada AES-128 diperlukan transformasi *cipher* dengan cara dibalik sehingga menghasilkan *inverse cipher* dengan tahapan yaitu: *InvShiftRows*, *InvSubBytes*, *InvMixColumns*, dan *AddRoundKey*. Algoritma dekripsi dapat dilihat pada gambar 2 berikut ini:



Gambar 2. Proses Dekripsi AES-128

Secara garis besar proses dekripsi AES-128 dengan kunci 128 bit adalah sebagai berikut [10]:

a. *InvShiftRows*: melakukan pergeseran *bit* ke kanan pada setiap blok baris.
b. *InSubBytes*: Setiap elemen pada *state* dipetakan dengan tabel *Inverse* S-Box.

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1x	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2x	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3x	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4x	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5x	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6x	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7x	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8x	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9x	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
ax	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
bx	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
cx	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
dx	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
ex	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
fx	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Gambar 3. Table *Inverse S-Box*

c. *InvMixColumns*: Setiap kolom dalam *state* dikalikan dengan matriks AES.
d. *AddRoundKey*: Mengombinasikan *state array* dan *round key* dengan hubungan XOR.
e. Pada proses terakhir akan menghasilkan karakter atau teks asli (*plaintext*).

3. DESAIN PENELITIAN

3.1 Bahan dan Alat Penelitian

Bahan penelitian yang digunakan yaitu halaman dari website LPPM UNIDAYAN yang memiliki URL yang belum dilakukan enkripsi. Untuk alat yang digunakan yaitu PHP dan MySQL serta untuk alat yang digunakan untuk pengujian keamanan menggunakan *SQLMap*.

3.2 Prosedur Penelitian

Untuk prosedur penelitian dapat dilihat pada gambar 4 di bawah ini.

Gambar 4. Prosedur Penelitian

4. HASIL DAN PEMBAHASAN

4.1 Analisis Keamanan Data

Langkah-langkah dalam pengelolaan keamanan pada sebuah website dapat dilihat dari kontribusi terhadap resiko antara lain:

a. Database

Untuk *database* yang rentan terhadap serangan yaitu pada tabel informasi dan tabel *login* seperti gambar 5 dan 6 di bawah ini.

Id_Info	Judul	Kategori	Informasi	Tanggal_Post	Waktu_Post	Status
1	Softfile Skripsi/Tesis di LPPM Unidayan	Kemahasiswaan	<p style="text-align:justify">Bagi mahasiswa yang ...	2019-10-01	11:00:00	Publish
25	Proposal Penelitian Mahasiswa di LPPM	Kemahasiswaan	<p>Surat Izin Penelitian dan Proposal Penelit...	2019-10-19	10:49:00	Publish

Gambar 5. Tabel Informasi

Id_Admin	Nama_Pengguna	Kata_Sandi	Hak_Akses
1	admin	c2068264bb0d1c9837dd99a83f8096a9	Admin

Gambar 6. Tabel *Login*

Berdasarkan Gambar 5 tersebut terdapat informasi yang berstatus *publish* yang berarti informasi tersebut dapat dilihat oleh orang lain (selain *Admin*) berdasarkan Id_Info yang terlihat pada URL yang dapat digunakan untuk melakukan serangan SQL *Injection* untuk mendapatkan informasi *login* akun dari *database* dengan mencari berdasarkan informasi dari beberapa tabel yang dihasilkan dari *SQLmap*.

b. Ancaman (*Threats*)

Terdapat beberapa ancaman keamanan dari sebuah sebuah website antara lain:

1) Ancaman dari pengguna (*Users*)

Ancaman yang berasal dari pengguna web yang biasanya dikarenakan kesalahan dari pengguna website tersebut, seperti dengan secara sengaja ataupun tidak secara sengaja memberitahukan *password* yang digunakan untuk *login* ke dalam sistem kepada orang yang tidak berhak.

2) Ancaman dari luar sistem

Ancaman keamanan dari luar sistem dapat berasal dari *hacker* yang berusaha dengan sengaja menyerang kelemahan yang terdapat pada suatu website untuk mendapatkan, merusak dan memanipulasi data yang ada.

4.2 Analisis Keamanan Data

a. Implementasi proses enkripsi

Simulasi Kriptografi AES-128 yang diterapkan pada URL dapat dilihat pada gambar 7 sebagai berkut:

URL Sebelum Enkripsi : https://www.lppmunidayan.ac.id/page.php?p=1

***Plaintext*:** p dan 1

***Random Key*:** 5b7e151628aed2a3abf7158809cf4f3d

***Chippertext*:** 5a4e57bf0396318dbc4198946f414b1d dan f7e1d7e5f03a4da58b48b32d08e92ba1

Secara Keseluruhan URL Setelah Enkripsi Sebagai Berikut:

https://www.lppmunidayan.ac.id/page.php?5a4e57bf0396318dbc4198946f414b1d=f7e1d7e5f03a4da58b48b32d08e92ba1

Gambar 7. Simulasi Enkripsi pada URL Website

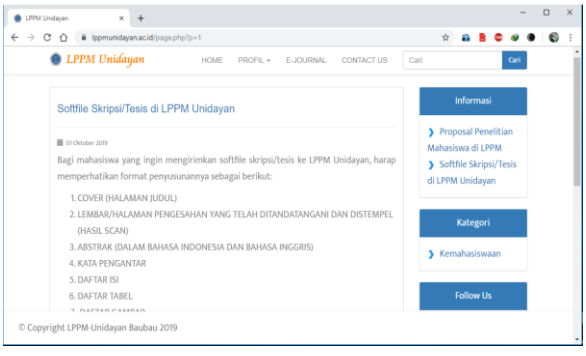
Pada simulasi tersebut terdapat URL yang telah dienkripsi secara keseluruhan menjadi https://www.lppmunidayan.ac.id/page.php?5a4e57bf0396318dbc4198946f414b1d=f7e1d7e5f03a4da58b48b32d08e92ba1. Proses enkripsi dengan AES-128 memerlukan suatu *Random Key* yang digunakan yaitu "5b7e151628aed2a3abf7158809 cf4f3d". Untuk karakter yang dienkripsi adalah hanya parameter *GET* URL saja yaitu karakter/huruf "p" dan angka "1". Angka 1 tersebut diperoleh dari databse dengan *filed* "Id_Info". Bagian parameter *GET* pada URL perlu dienkripsi, karena pada bagian tersebut terdapat informasi atau isi dari basis data dan dapat menjadi celah bagi para *hacker* untuk mengakses data-data lain yang bersifat rahasia.

b. Pengujian Keamanan Data

Untuk pengujian keamanan data dilakukan dengan beberapa tahapan serangan *SQL Injection* yaitu dengan menggunakan aplikasi *SQLmap*. Proses serangan *SQL Injection* dilakukan sebelum URL dilakukan enkripsi dan sesudah dilakukan enkripsi.

Berikut merupakan tahapan serangan *SQL injection* sebelum dilakukan enkripsi:

1) Menentukan link URL yang akan diserang



Gambar 8. Web LPPM Unidayan sebelum URL dienkripsi

Berdasarkan gambar 8 tersebut diperoleh URL https://www.lppmunidayan.ac.id/page.php?p=1.

2) Memasukkan perintah *SQL Injection* untuk melihat daftar database.

```
C:\sqlmap_1.3>sqlmap.py -u https://www.lppmunidayan.ac.id/page.php?p=1 --dbs
```

Gambar 9. *SQL Injection* menampilkan daftar database

Berdasarkan perintah pada Gambar 9, maka diperoleh beberapa database yang ditemukan sebagai berikut.

```
[20:56:57] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.0.12
[20:56:57] [INFO] fetching database names
[20:57:05] [INFO] used SQL query returns 2 entries
[20:57:07] [INFO] retrieved: 'information_schema'
[20:57:08] [INFO] retrieved: 'lppmunid_lppm11'
available databases [2]:
[*] information_schema
[*] lppmunid_lppm11
```

Gambar 10. Kumpulan database yang ditemukan dari perintah *SQL Injection*

3) Memasukkan perintah *SQL Injection* untuk melihat daftar tabel di dalam database.

Berdasarkan gambar 10, selanjutnya dipilih database yang ingin dilihat daftar tabelnya yaitu dengan nama database "lppmunid_lppm11".

Untuk melihat isi atau daftar tabel dari databse tersebut digunakan perintah sebagai berikut.

```
C:\sqlmap_1.3>sqlmap.py -u https://www.lppmunidayan.ac.id/page.php?p=1 lppmunid_lppm11 --tables
```

Gambar 11. *SQL Injection* menampilkan daftar tabel

Kemudian akan menampilkan daftar tabel sebagai berikut

```
Database: lppmunid_lppm11
[6 tables]
+---------------------------------------+
| contact                               |
| informasi                             |
| kategori                              |
| login                                 |
| struktur                              |
| visimisi                              |
+---------------------------------------+
```

Gambar 12. Kumpulan tabel yang ditemukan dari perintah *SQL Injection*

4) Melihat isi baris dari salah satu tabel

Berdasarkan gambar 12 tersebut terdapat tabel *login* yang berisikan informasi *login* ke halaman *admin* pada Web LPPM Undayan, untuk itu langkah terakhir adalah melihat isi baris dari tabel *login* tersebut dengan perintah sebagai berikut:

```
C:\sqlmap_1.3>sqlmap.py -u https://www.lppmunidayan.ac.id/page.php?p=1 lppmunid_lppm11 -t login
--dump
```

Gambar 13. *SQL Injection* melihat isi baris dari tabel *login*

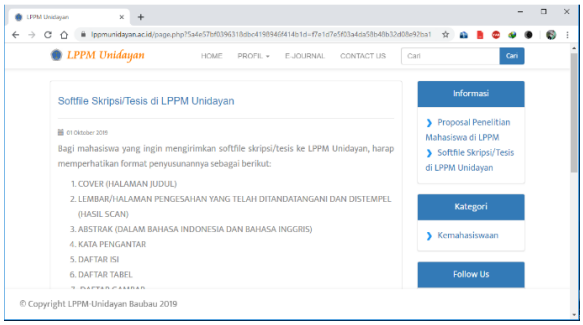
Berdasarkan perintah pada gambar 13 tersebut maka akan menghasilkan informasi baris dari tabel *login* sebagai berikut:

```
Database: lppmunid_lppm11
Table: login
[1 entry]
+----------+-----------+----------------------------------+--------------+
| Id_Admin | Hak_Akses | Kata_Sandi                       | Nama_Pengguna |
+----------+-----------+----------------------------------+--------------+
| 1        | Admin     | 7efc460df2a0ac57184311cd529c64c4 | admin@admin   |
+----------+-----------+----------------------------------+--------------+
```

Gambar 14. Isi baris dari tabel *login*

Berdasarkan informasi dari gambar 14 tersebut diperoleh Nama Pengguna dan Kata Sandi untuk dapat masuk ke halaman *Admin* website LPPM unidayan yang dapat disalahgunakan oleh orang-orang yang tidak bertanggung jawab. Untuk itu diperlukan sebuah enkripsi URL untuk mencegah serangan *SQL Injection*. Berikut merupakan pengujian setelah penerapan Enkripsi dengan kriptografi AES-128.

a) Menentukan URL yang akan diserang setelah penerapan enkripsi



Gambar 15. Web LPPM Unidayan setelah dilakukan enkripsi URL

Berdasarkan gambar 15 tersebut diperoleh URL sebagai berikut: https://www.lppmunidayan.ac.id/page.php?5a4e57bf0396318dbc4198946f414b1d=f7e1d7e5f03a4da58b48b32d08e92ba1.

b) Memasukkan perintah SQL *Injection* untuk melihat daftar database.

```
C:\sqlmap_1.3>sqlmap.py -u https://www.lppmunidayan.ac.id/
page.php?5a4e57bf0396318dbc4198946f414b1d=f7e1d7e5f03a4da5
8b48b32d08e92ba1 --dbs
```

Gambar 16. Perintah *SQL Injection* untuk melihat daftar database dari link yang dienkripsi

Berdasarkan perintah pada gambar 16 tersebut, maka akan menampilkan informasi bahwa parameter yang digunakan yaitu "5a4e57bf0396318 dbc4198946f414b1d" tidak dapat dilakukan injeksi SQL, informasi tersebut dapat dilihat pada gambar 17 berikut.

```
[18:45:35] [WARNING] GET parameter '5a4e57bf0396318dbc4198
946f414b1d' does not seem to be injectable
[18:45:35] [CRITICAL] all tested parameters do not appear
to be injectable. Try to increase values for '--level'/'--
risk' options if you wish to perform more tests. If you su
spect that there is some kind of protection mechanism invo
lved (e.g. WAF) maybe you could try to use option '--tampe
r' (e.g. '--tamper=space2comment') and/or switch '--random
-agent'
```

Gambar 17. Hasil *SQL Injection* terhadap URL yang dienkripsi

Berdasarkan hasil pengujian yang telah dilakukan dapat diketahui bahwa sebelum diterapkannya kriptografi pada URL maka data-data berupa database maupun isi dari tabel pada database dapat terlihat dengan mudah menggunakan perintah *SQL injection* sedangkan setelah diterapkan Kriptografi AES-128 pada URL maka hasil dari *SQL Injection* akan menampilkan informasi bahwa tidak dapat melakukan Injeksi SQL.

5. KESIMPULAN

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan mengenai penerapan kriptografi AES-128 untuk mengamankan URL website dapat disimpulkan bahwa:

a. Dengan diterapkannya cara pengamanan tersebut, URL website dapat disamarkan, sehingga dapat mengatasi serangan yang mengancam keamanan data dalam suatu website.
b. Integritas dari URL yang telah dienkripsi akan lebih terjaga, karena metode SQL injection tidak dapat diterapkan pada URL yang telah dilakukan enkripsi.

DAFTAR REFERENSI

[1] M. Yuhefizar and R. Hidayat, *Cara Mudah Membangun Website Interaktif Menggunakan Content Management System Joomla Edisi Revisi*. Jakarta: PT Elex Media Komputindo, 2009.

[2] R. Ferreira and R. L. Aguiar, "Repositioning privacy concerns: Web servers controlling URL metadata," *J. Inf. Secur. Appl.*, vol. 46, pp. 121–137, 2019, doi: 10.1016/j.jisa.2019.03.010.

[3] P. C. Xue, "SQL injection attack and guard technical research," in *Procedia Engineering*, 2011, vol. 15, pp. 4131–4135, doi: 10.1016/j.proeng.2011.08.775.

[4] A. P. Bhatt and S. Anand, "Quantum Cryptography for Internet of Things Security," *J. Electron. Scicence Technol.*, vol. 17, no. 3, pp. 213–220, 2019, doi: 10.11989/JEST.1674-862X.90523016.

[5] N. B. F. Silva, D. F. Pigatto, P. S. Martins, and K. R. L. J. C. Branco, "Case studies of performance evaluation of cryptographic algorithms for an embedded system and a general purpose computer," *J. Netw. Comput. Appl.*, vol. 60, pp. 130–143, 2016, doi: 10.1016/j.jnca.2015.10.007.

[6] R. Munir, *Kriptografy*. Bandung: Informatika, 2006.

[7] D. Smekal, J. Frolka, and J. Hajny, "Acceleration of AES Encryption Algorithm Using Field Programmable Gate Arrays," in *IFAC-PapersOnLine*, 2016, vol. 49, no. 25, pp. 384–389, doi: 10.1016/j.ifacol.2016.12.075.

[8] P. Patil, P. Narayankar, D. G. Narayan, and S. M. Meena, "A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish," in *Procedia Computer Science*, 2016, vol. 78, no. December 2015, pp. 617–624, doi: 10.1016/j.procs.2016.02.108.

[9] D. S. Kundi, A. Aziz, and N. Ikram, "A high performance ST-Box based unified AES encryption/decryption architecture on FPGA," *Microprocess. Microsyst.*, vol. 41, pp. 37–46, 2016, doi: 10.1016/j.micpro.2015.11.015.

[10] J. M. Granado, M. A. Vega-Rodríguez, J. M. Sánchez-Pérez, and J. A. Gómez-Pulido, "IDEA and AES, two cryptographic algorithms implemented using partial and dynamic reconfiguration," *Microelectronics J.*, vol. 40, no. 6, pp. 1032–1040, 2009, doi: 10.1016/j.mejo.2008.11.044.