PENERAPAN KRIPTOGRAFI *BASE*64 UNTUK KEAMANAN URL (*UNIFORM RESOURCE LOCATOR*) *WEBSITE* DARI SERANGAN *SQL INJECTION*

Aziz Pratama Nugraha[1], Erwin Gunadhi[2]

Jurnal Algoritma
Sekolah Tinggi Teknologi Garut
Jl. Mayor Syamsu No. 1 Jayaraga Garut 44151 Indonesia
Email : jurnal@sttgarut.ac.id

[1]1206021@sttgarut.ac.id
[2]erwin.gunadhi@sttgarut.ac.id

Abstrak *– URL menunjukan alamat dari sebuah homepage atau menunjukan sumber daya Internet, yaitu alamat suatu dokumen atau program yang ingin ditampilkan atau digunakan. URL dapat menjadi salah satu aspek yang menjadi kelemahan pada suatu website, karena pada URL terdapat berbagai informasi yang berisikan protocol, alamat server dan path file yang dapat digunakan untuk melakukan aksi SQL Injection. Salah satu cara yang dapat digunakan untuk mengamankan suatu website dari serangan SQL injection adalah dengan ilmu kriptografi. Ilmu kriptografi dapat menyamarkan URL website menjadi kode atau sandi yang tidak dapat dibaca oleh sembarang orang, sehingga dapat mencegah serangan SQL injection pada suatu website. Penerapan keamanan URL website dari serangan SQL injection ini menggunakan metode kriptografi dengan algoritma base64. Sedangkan untuk pemodelan data menggunakan flowchart. HTML, PHP, CSS, dan Javascript adalah bahasa pemrograman yang digunakan dalam pembangunan perangkat lunak, serta Sublime Text 2 digunakan sebagai media penulisan script dari bahasa pemrograman tersebut. Basis data yang digunakan adalah MySQL yang terintegrasi dalam aplikasi XAMPP dan Mozilla Firefox sebagai media browser. Untuk melakukan pengujian terhadap keamanan website dari serangan SQL injection menggunakan aplikasi Web Cruiser Web Vulnerability Scanner.Dengan diterapkannya cara pengamanan dengan kriptografi base64, URL website dapat disamarkan. Hal tersebut dapat mengatasi serangan yang mengancam keamanan data pada suatu website. Integritas dari URL yang telah dienkripsi akan lebih terjaga, karena metode SQL injection tidak dapat diterapkan pada URL yang telah dienkripsi.*

Kata Kunci *–* *Base*64, Keamanan, Kriptografi, *SQL Injection*, URL

I. PENDAHULUAN

Website merupakan sarana yang digunakan untuk menyebarkan informasi melalui Internet, baik berupa teks, gambar, suara, maupun video. Seiring penggunaan *website* yang semakin luas, dapat menimbulkan berbagai macam tindak kejahatan seperti pencurian, manipulasi data atau informasi penting dari suatu *website* oleh orang yang tidak bertanggung jawab.

Dokumen-dokumen informasi yang terdapat pada website dihubungkan melalui alamat URL (Uniform Resource Locator). Di dalam URL terdapat alamat server, lokasi dan nama dokumen yang terdapat pada suatu website. 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, cara yang banyak digunakan adalah dengan menggunakan metode SQL injection. SQL injection menjadi sangat terkenal di Indonesia semenjak dibobolnya situs KPU pada Pemilu 2004 lalu [7].

Metode *SQL injection* digunakan untuk memasukan perintah SQL sebagai *input* pada suatu *website* untuk mendapatkan akses ke dalam basis data. Jika basis data *website* dapat diakses, maka seorang *hacker* dapat dengan mudah mencuri berbagai data rahasia, bahkan dapat memanupulasi atau merusak data pada *website* tersebut. Salah satu cara yang dapat digunakan untuk mengamankan suatu *website* dari serangan *SQL injection* adalah dengan ilmu kriptografi. Dengan ilmu kriptografi tersebut, URL pada *website* dapat disamarkan menjadi kode atau sandi yang tidak dapat dibaca oleh sembarang orang, sehingga dapat mencegah serangan *SQL injection* pada suatu *website*.

II. TINJAUAN PUSTAKA

A. Keamanan

Masalah keamanan salah satu aspek penting dalam suatu sistem, tetapi sering kali keamanan suatu sistem dikesampingkan. Sering kali pada perancangan suatu sistem, keamanan berada diurutan setelah tampilan atau bahkan berada diurutan terakhir dari daftar hal-hal yang dianggap penting dari suatu sistem. Apabila mengganggu performansi dari sistem, seringkali masalah keamanan tidak begitu dipedulikan, bahkan ditiadakan [1]. Pengelolaan terhadap keamanan dapat dilihat dari sisi pengelolaan resiko (*risk management*). Lawrie Brown dalam Lee (2000) yang dikutip oleh Rahardjo [4] menyarankan menggunakan "*Risk Management Model*" untuk menghadapi ancaman (*managingthreats*). *Asset*, *Vulnerabilities*, dan *Threats* merupakan tiga komponen yang memberikan kontribusi terhadap *risk*.

Ancaman keamanan yang dapat terjadi terhadap informasi [1], *Interruption*, *Interception*, Modifikasi, *Fabrication*. Sedangkan terdapat beberapa aspek keamanan yang diperlukan agar dapat terhindar dari tindakan kejahatan [2], diantaranya aspek *authentication*, aspek *integrity*, aspek *non-repudiation*, aspek *authority*, aspek *confidentiality*, aspek *privacy*, aspek *availability*, aspek *access control*.

B. Kriptografi

Arti kata kriptografi berasal dari bahasa yunani, yaitu *kripto* dan *graphia*. Arti dari kata *Kripto* adalah *secret* (rahasia) dan arti dari kata *graphia* adalah *writing* (tulisan). Berdasarkan dari terminologinya, kriptografi merupakan ilmu dan seni yang digunakan untuk menjaga keamanan pesan yang dikirim dari suatu tempat ke tempat yang lain [1].

Algoritma kriptografi terdiri dari tiga fungsi dasar [1], diantaranya enkripsi, dekripsi, dan kunci. Algoritma dapat dibagi menjadi tiga bagian berdasarkan kunci yang dipakainya [1] yaitu algoritma simetri, algoritma asimetri, *hash function*.

C. Algoritma *Base*64

Transformasi *base*64 merupakan salah satu algoritma untuk *encoding* dan *decoding* suatu data ke dalam format ASCII, yang didasarkan pada bilangan dasar 64 atau bisa dikatakan sebagai salah satu metoda yang digunakan untuk melakukan *encoding* (penyandian) terhadap data *binary*. Karakter yang dihasilkan pada transformasi *base*64 ini terdiri dari A..Z, a..z dan 0..9, serta ditambah simbol "+" dan "/" serta satu buah karakter sama dengan (=) di dua karakter terakhir yang dipakai untuk pengisian *pad* atau dengan kata lain penyesuaian dan menggenapkan data *binary*. Karakter simbol yang akan dihasilkan akan tergantung dari proses algoritma yang berjalan. [6]

Kriptografi transformasi *base*64 banyak digunakan di dunia Internet sebagai media data format untuk mengirimkan data, penggunaan tersebut dikarenakan hasil dari *encode base*64 berupa *plaintext*, maka data ini akan jauh lebih mudah dikirim, dibandingkan dengan format data yang berupa *binary*. Algoritma *base*64 menggunakan kode ASCII dan kode *index base*64 dalam melakukan proses enkripsi ataupun dekripsinya. Dalam melakukan enkripsi pada URL *website*, kode *index base*64 perlu dimodifikasi. Simbol "+" dimodifikasi menjadi "-" dan simbol simbol "/" menjadi "_". Tabel index *base*64 dapat dilihat pada tabel 1.

Tabel 1 : Kode *Index Base*64 (URL *and Filename Safe*) [3]

Value	*Encoding*	*Value*	*Encoding*	*Value*	*Encoding*	*Value*	*Encoding*
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	-
15	P	31	f	47	v	63	_
						(*pad*)	=

Menurut Ariyus (2008) yang dikutip oleh Wahyu [6], teknik enkripsi *base*64 sebetulnya sederhana, jika terdapat sebuah (*string*) *bytes* yang akan disandikan ke algoritma *base*64 maka tahapannya yaitu:

1. Pecah *string bytes* tersebut ke per-3 *bytes*.
2. Gabungkan 3 *bytes* menjadi 24 *bit*. dengan catatan 1 *bytes* = 8 *bit*, sehingga 3 x 8 = 24 *bit*.
3. Lalu 24 *bit* yang disimpan di-*buffer* (disatukan) dipecah-pecah menjadi 6 *bit*, maka akan menghasilkan 4 pecahan.
4. Masing masing pecahan diubah ke dalam nilai desimal, dimana maksimal nilai 6 *bit* dalah 63.
5. Terakhir, jadikan nilai-nilai desimal tersebut menjadi *index* untuk memilih maksimal *index* ke 64 atau karakter ke 63 dari penyusun *base*64.

Dan seterusnya hingga akhir *string bytes* yang akan mengalami konversi. Apabila dalam proses *encoding* terdapat sisa pembagi, maka tambahkan karakter *pad* (=) sebagai penggenap sisa tersebut. Oleh karena itu, terkadang pada *base*64 akan muncul satu atau dua karakter (=).

D. URL *(Uniform Resource Locator)*

URL menunjukan alamat dari sebuah *homepage* atau menunjukan sumber daya Internet, yaitu alamat suatu dokumen atau program yang ingin ditampilkan atau digunakan. Bagian pertama URL menunjukan *protocol*, misalnya *http://* atau *https://*. *Protocol* merupakan persetujuan bersama yang digunakan untuk melakukan komunikasi, dalam hal ini menggunakan HTTP (*Hypertext Transfer Protocol*). Bagian kedua dari URL menunjukan alamat *server* tempat disimpannya sumber daya tersebut, misalnya *www.microsoft.com* untuk *website Microsoft Corporation*. Selanjutnya untuk bagian ketiga dari URL adalah *path file*, merupakan bagian dari URL yang menunjukan lokasi dan nama dokumen atau *program* dalam *server* tersebut. [5]

E. Definisi *SQL Injection*

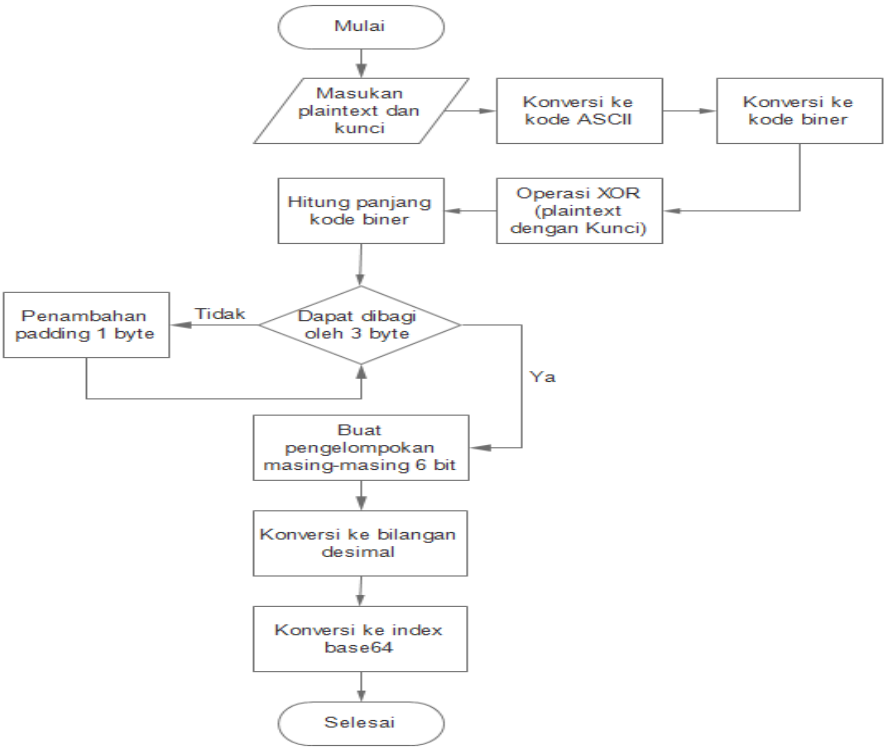
SQL injection terdiri dari dua kata, yaitu SQL merupakan sebuah bahasa yang digunakan untuk mengakses suatu basis data, sedangkan kata *injection* jika diterjemahkan memiliki arti menyuntik. *SQL injection* adalah sebuah metode untuk memasukan perintah SQL sebagai *input* melalui sebuah *web* guna mendapatkan akses *database* [7].

III. KERANGKA KERJA KONSEPTUAL

Dalam melakukan proses enkripsi dan dekripsi pada kriptografi *base*64 diperlukan beberapa tahapan penyelesaian yang dapat digambarkan dalam bentuk *flowchart*.

1. Flowchart Enkripsi

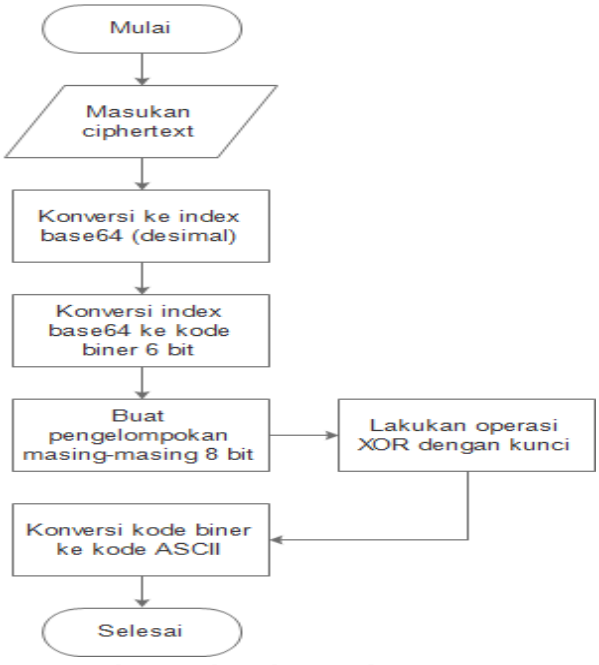
Flowchart enkripsi algoritma *base*64 menjelaskan alur dari proses enkripsi yang akan diterapkan untuk mengamankan URL dari serangan *SQL injection*.



Gambar 1 *Flowchart* Enkripsi *Base*64

2. Flowchart Dekripsi

Flowchart dekripsi algoritma *base*64 menjelaskan alur dari proses dekripsi yang akan diterapkan untuk mengamankan URL dari serangan *SQL injection*.



Gambar 2 *Flowchart* Dekripsi *Base*64

IV. HASIL DAN PEMBAHASAN

A. Analisis Keamanan Data

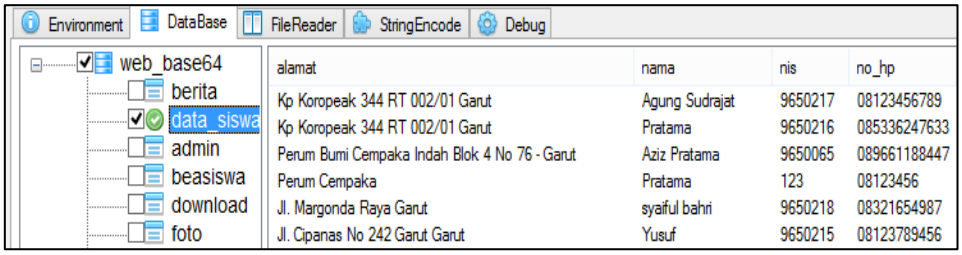
Langkah-langkah pengelolaan keamanan pada sebuah sistem dapat dilihat dari kontribusi terhadap resiko, diantaranya:

1. Aset (*assets*)

   Terdapat berbagai macam aset pada suatu *website* yang harus dilindungi, diantaranya:

   a. Data Pengguna

      SQL injection dapat mengakses data pengguna pada suatu *website*, seperti yang dapat dilihat pada gambar 3.

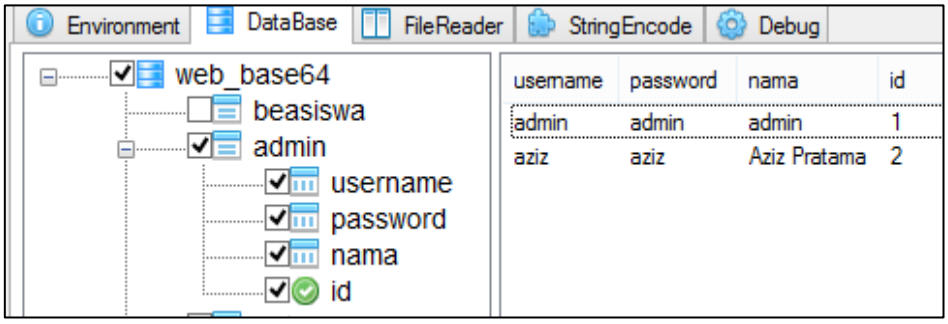


Gambar 3 SQL Injection Data Pengguna

Data pengguna pada suatu *website* seharusnya menjadi data rahasia yang tidak boleh diketahui sembarang orang, karena dapat digunakan untuk melakukan tindakan kejahatan. Dimisalkan data nama dan nomor telepon yang sering digunakan dalam melakukan aksi penipuan.

   b. Akun yang digunakan untuk *login* ke dalam *website*

      Serangan *SQL injection* dapat digunakan untuk mengetahui data yang digunakan untuk *log-in* ke dalam suatu *website*, seperti yang dapat dilihat pada gambar 4.



Gambar 4 *SQL Injection* Akun *Website*

Akun pada suatu *website* seharusnya menjadi data rahasia yang tidak boleh diketahui sembarang orang, karena dapat digunakan untuk melakukan tindakan kejahatan. Dimisalkan *username* dan *password* yang digunakan untuk *login* diketahui oleh orang yang tidak bertanggung jawab, maka data pada *website* tersebut dapat dicuri, diubah, dihapus atau bahkan *website* tersebut dapat diambil alih. *Website* dapat diambil alih dengan cara mengganti data *username* atau *password* yang digunakan untuk *login*, sehingga pemiliknya tidak dapat masuk ke dalam *website* tersebut.

2. Ancaman (*threats*)

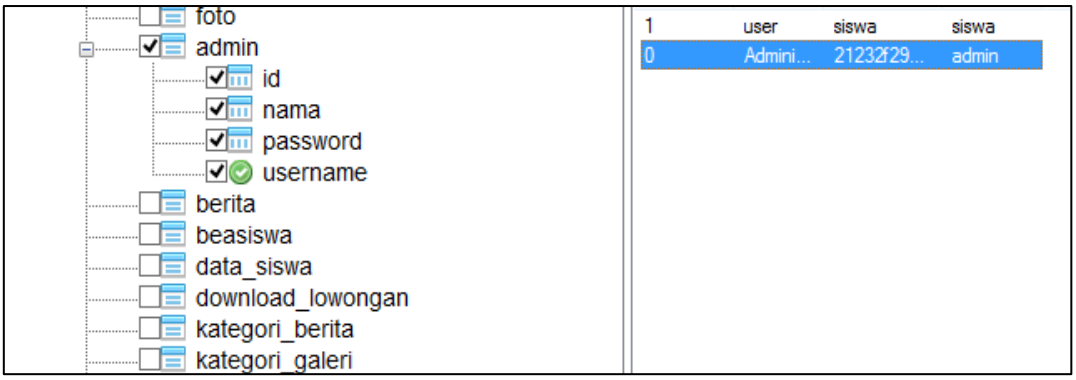
   Terdapat berbagai macam aspek yang dapat mengancam keamanan dari suatu *website*, diantaranya:

   a. Ancaman yang berasal dari pengguna (*users*)

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

b. Ancaman yang berasal dari luar sistem
Ancaman keamanan dari luar sistem dapat berasal dari orang yang tidak bertanggung jawab, seperti *hacker* yang dengan sengaja menyerang kelemahan yang terdapat pada suatu *website*, untuk mendapatkan, memanipulasi ataupun merusak data yang ada.

3. Kelemahan (*Vulnerabilities*)
URL dapat menjadi salah satu aspek yang menjadi kelemahan pada suatu *website*, karena pada URL terdapat berbagai informasi yang berisikan *protocol*, alamat *server* dan *path file*. Hanya dengan mencari kelemahan dengan memanfaatkan informasi tersebut dapat dengan mudah dimanfaatkan oleh *hacker* untuk dapat mengakses basis data pada suatu *website*. Jika basis data pada suatu *website* dapat diakses, maka para *hacker* dapat dengan mudah mencuri, mengubah ataupun menghapus data pada *website* tersebut. Sebagai contoh, basis data yang dapat diakses dengan menggunakan *SQL injection* dapat dilihat pada gambar 5.



Gambar 5 Basis Data dapat Diakses dengan *SQL Injection*

B. Perancangan Keamanan Data

Kelemahan pada algoritma *base*64 terdapat pada kode *index* yang telah diketahui susunan datanya. Untuk mengatasi kelemahan tersebut, pada penelitian ini susunan dari tabel *index* tersebut akan diubah sehingga menjadi seperti yang dapat dilihat pada tabel 2.

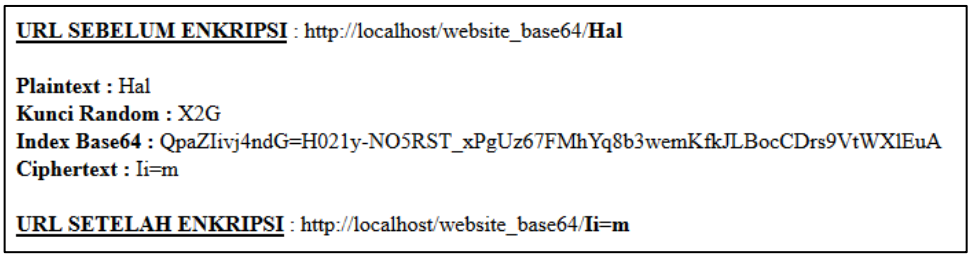
Tabel 2 : Kustomisasi Kode *Index Base*64

Value	*Encoding*	*Value*	*Encoding*	*Value*	*Encoding*	*Value*	*Encoding*
0	Q	16	1	32	7	48	L
1	p	17	y	33	F	49	B
2	a	18	-	34	M	50	o
3	Z	19	N	35	h	51	c
4	I	20	O	36	Y	52	C
5	i	21	5	37	q	53	D
6	v	22	R	38	8	54	r
7	j	23	S	39	b	55	s
8	4	24	T	40	3	56	9
9	n	25	_	41	w	57	V
10	d	26	x	42	e	58	t
11	G	27	P	43	m	59	W
12	=	28	g	44	K	60	X
13	H	29	U	45	f	61	l
14	0	30	z	46	k	62	E
15	2	31	6	47	J	63	u
						(*pad*)	A

C. Penerapan dan Pengujian

1. Implementasi Proses Enkripsi

Algoritma kriptografi *base*64 yang telah dikustomisasi diterapkan pada URL dengan menggunakan bahasa pemrograman PHP. Simulasi penerapan kriptografi *base*64 yang telah dikustomisasi pada URL *website* dapat dilihat pada gambar 6.



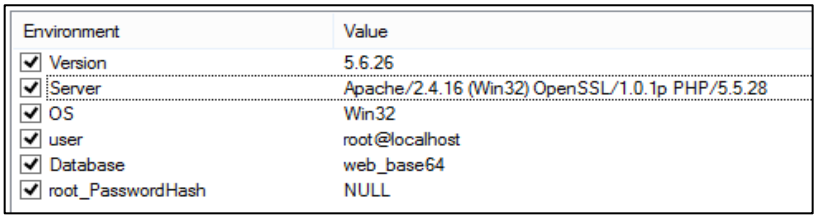
Gambar 6 Simulasi Base64 pada URL Website

Pada simulasi tersebut terdapat URL "*http://localhost/website_base64/Hal*", dengan kunci random "X2G". URL yang dienkripsi hanya pada nilai parameter dari *path file* atau pada bagian yang dapat mengidentifikasikan atribut yang berhubungan dengan data saja, dalam simulasi tersebut yang menjadi *path file* adalah kata "Hal". Bagian *path file* pada URL perlu dienkripsi, karena pada bagian tersebut terdapat informasi yang memberitahukan lokasi penyimpanan data pada *storage* ataupun basis data dan dapat menjadi celah bagi para *hacker* untuk mengakses data-data lain yang bersifat rahasia.

2. Pengujian Keamanan Data

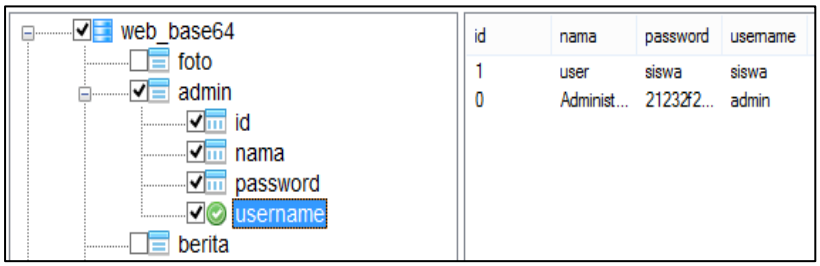
Pada sub bab ini akan dilakukan pengujian keamanan dengan melakukan serangan terhadap URL *website* dengan metode *SQL injection*. *SQL injection tool* yang digunakan dalam pengujian ini adalah Web Cruiser Web Vulnerability Scanner. Proses pengujian akan dibagi menjadi dua bagian, yaitu pengujian *website* sebelum dan setelah diterapkan kriptografi *base*64.

a. Berdasarkan pengujian yang telah dilakukan terhadap URL sebelum diterapkan kriptografi *base*64, didapatkan informasi seputar *server* dan basis data yang digunakan. Seperti yang dapat dilihat pada gambar 7.



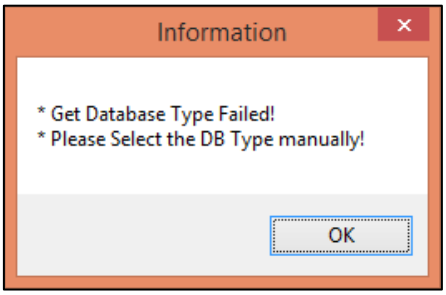
Gambar 7 Informasi Sebelum Diterapkan Kriptografi

Informasi tersebut kemudian dapat digunakan untuk mengakses data yang telah tersimpan pada basis data *website*. Seperti yang dapat dilihat pada gambar 8.



Gambar 8 Basis Data *Website*

b. Berdasarkan pengujian yang telah dilakukan terhadap URL setelah diterapkan kriptografi *base*64, informasi *server* dan basis data tidak dapat ditemukan. Seperti yang dapat dilihat pada gambar 9.



Gambar 9 Informasi Server dan Basis Data Tidak Dapat Ditemukan

Berdasarkan dari hasil pengujian yang telah dilakukan, diketahui bahwa sebelum diterapkannya kriptografi pada URL *website*, basis data dapat diakses dengan mudah menggunakan *SQL injection*. Sedangkan dengan diterapkannya kriptografi *base*64 pada URL *website*, informasi *server* dan basis data tidak dapat diakses. Hal tersebut dapat mengatasi ancaman terhadap keamanan data, terutama dari serangan *SQL injection*.

V. KESIMPULAN

Berdasarkan berbagai penjelasan dan hasil penelitian yang telah dilakukan, mengenai cara mengamankan URL *website* dari serangan *SQL injection*. Dapat disimpulkan beberapa hal, diantaranya:

1. Dengan diterapkannya cara pengamanan ini, URL *website* dapat disamarkan. Hal tersebut dapat mengatasi serangan yang mengancam keamanan data pada suatu *website*.
2. Integritas dari URL yang telah dienkripsi akan lebih terjaga, karena metode *SQL injection* tidak dapat diterapkan pada URL yang telah dienkripsi.

UCAPAN TERIMAKASIH

Penulis mengucapkan banyak terimakasih kepada ibu, ayah, adik dan kawan-kawan yang telah memberikan dukungan moril maupun materil kepada penulis. Penulis juga menyampaikan terimakasih kepada Bapak Rd. Erwin Gunadhi, Ir., MT., selaku pembimbing yang telah memberikan arahan serta bimbingan selama proses penyelesaian laporan tugas akhir ini.

DAFTAR PUSTAKA

[1] Ariyus, D. (2006). *Kriptografi.* Yogyakarta: Graha Ilmu.
[2] Ariyus, D. (2009). *Keamanan Multimedia.* Yogyakarta: Andi Offset.
[3] Josefsson, S. (2006). *The Base16, Base32, and Base64 Data Encodings.* Retrieved Juni 11, 2016, from IETF Tools web site on World Wide Web: https://tools.ietf.org/pdf/draft-josefsson-rfc3548bis-04.pdf
[4] Rahardjo, B. (2005). *Keamanan Sistem Informasi Berbasis Internet.* Jakarta: PT INDOCISC.
[5] Sunarto. (2008). *Teknologi Informasi dan Komunikasi.* Jakarta: Grasindo.
[6] Wahyu, F., Rahangiar, A. P., & Fretes, F. d. (2012). *Penerapan Algoritma Gabungan RC4 dan Base64 Pada Sistem Keamanan E-Commerce.* Retrieved Mei 25, 2016, from Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana web site on World Wide Web: http://journal.uii.ac.id/index.php/Snati/article/viewFile/2873/2628
[7] Zam, E. (2012). *SQL Injection.* Jakarta: Elex Media Komputindo.