MENGAMANKAN SKRIP PADA BAHASA PEMOGRAMAN PHP DENGAN MENGGUNAKAN KRIPTOGRAFI BASE64

Ahmad Timbul Sholeh[1], Erwin Gunadhi[2], Asep Deddy Supriatna[3]

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

[1]0906018@sttgarut.ac.id
[2]erwingunadhi@sttgarut.ac.id
[3]asepdeddy@sttgarut.ac.id

Abstrak *– Perkembangan teknologi informasi saat ini mengharuskan setiap perusahaan untuk dapat meningkatkan kualitas kinerjanya dalam upaya menghadapi persaingan global yang semakin pesat. Perusahaan dan pelanggan tidak lagi dibatasi oleh jarak dan waktu dimana semuanya dilakukan melalui media website dalam internet. Untuk merancang sebuah website, seorang programer dapat menggunakan beberapa bahasa program salah satunya adalah PHP (Hypertext Prepocessor). Akan tetapi, perangkat lunak hasil PHP (Hypertext Prepocessor) harus didistribusikan dalam bentuk source, sehingga memiliki beberapa kekurangan dan celah keamanan. Beberapa kekurangan tersebut salah satunya adalah skrip dapat dengan mudah disalin, diubah, ataupun digunakan sebagian/ seluruh dalam perangkat lunak lainnya tanpa ada pemberitahuan. Selain itu skrip yang tidak terenkripsi membuat perangkat lunak yang dibangun sangat rentan, karena skrip dapat mengungkapkan beberapa kelemahan dari perangkat lunak tersebut. Oleh karena itu, Penelitian ini bertujuan untuk mengamankan skrip dari PHP (Hypertext Prepocessor) yang akan didistribusikan supaya terjaga hak akses dan integritasnya.*
Penelitian menggunakan algoritma base64 dengan mengubah struktur index-nya yang bertujuan untuk menghamburkan makna dari plainteks ketika cipherteks dicoba untuk dipecahkan oleh pemecah kode. Pemodelan data menggunakan flowchart dan dalam implementasi menggunakan bahasa pemograman C#.
Dengan adanya cara pengamanan ini, pengembang aplikasi yang menggunakan bahasa pemograman PHP dapat menyembunyikan skrip php supaya tidak mudah disalin, diubah sebagian/ seluruhnya oleh orang yang tidak berhak dan dapat mengamankan kelemahan dari alur program aplikasi php.

Kata Kunci - *keamanan, kriptografi, base64, skrip, php*

1. PENDAHULUAN

Perkembangan teknologi informasi saat ini mengharuskan setiap perusahaan untuk dapat meningkatkan kualitas kinerjanya dalam upaya menghadapi persaingan global yang semakin pesat. Perusahaan dan pelanggan tidak lagi dibatasi oleh jarak dan waktu dimana semuanya dilakukan melalui media *website* dalam internet. Seiring penggunaannya yang semakin luas menimbulkan sebuah kejahatan yang disebut dengan "*cybercrime*". Kejahatan tersebut seperti mencuri PIN kartu kredit, menyadap data pribadi seseorang seperti alamat email, atau memanipulasi informasi dari sebuah halaman website dengan tujuan untuk mendapatkan apa yang diinginkannya.

Untuk merancang sebuah *website*, seorang programer dapat menggunakan beberapa bahasa program salah satunya adalah PHP (*Hypertext Prepocessor*). Menurut Builtwith [2], pada saat ini PHP (*Hypertext Prepocessor*) digunakan pada lebih dari 22 juta situs web dan PHP (*Hypertext Prepocessor*) semakin stabil dan lebih diperluas dari versi-versi pendahulunya. *Website* yang

menggunakan bahasa pemograman PHP antara lain untuk kebutuhan bisnis, teknologi, belanja, berita, edukasi, sosial, hiburan, dll

Akan tetapi, perangkat lunak hasil PHP (*Hypertext Prepocessor*) harus didistribusikan dalam bentuk *source*, sehingga memiliki beberapa kekurangan dan celah keamanan. Beberapa kekurangan tersebut salah satunya adalah skrip dapat dengan mudah disalin, diubah, ataupun digunakan sebagian dalam perangkat lunak lainnya tanpa ada pemberitahuan. Selain itu skrip yang tidak terenkripsi membuat perangkat lunak yang dibangun sangat rentan, karena *skrip* dapat mengungkapkan beberapa kelemahan dari perangkat lunak tersebut. Karena itulah diperlukan adanya suatu solusi yang dapat mengamankan *skrip* aplikasi PHP (*Hypertext Prepocessor*) yang akan didistribusikan.

2. LANDASAN TEORI

A. Keamanan

Masalah keamanan merupakan salah satu aspek penting dari sebuah sistem informasi. sayang sekali masalah keamanan ini sering kali kurang mendapat perhatian dari para pemilik dan pengelola sistem informasi. Seringkali masalah keamanan berada di urutan kedua, atau bahkan di urutan terakhir dalam daftar hal-hal yang dianggap penting. Apabila menggangu performansi dari sistem, seringkali keamanan dikurangi atau ditiadakan (Dowd & McHenry, 1998: 24-28) yang dikutip oleh Rahardjo [5]. Pengelolaan terhadap keamanan dapat di lihat dari sisi pengelolaan resiko (risk management). Lawrie Brown dalam (Lee, 2000) yang dikutip oleh Rahardjo [5] menyarankan menggunakan "*Risk Management Model*" untuk menghadapi ancaman (*managing threats*). Ada tiga komponen yang memberikan kontribusi kepada Risk, yaitu *Asset*, *Vulnerabilities*, dan *Threats*.

Garfinkel mengemukakan bahwa keamanan komputer *(computer security*) melingkupi empat aspek, yaitu *privacy*, *integrity*, *authentication*, dan *availability*. Selain keempat hal di atas, masih ada dua aspek lain yang juga sering dibahas dalam kaitannya dengan *electronic commerce*, yaitu access control dan *non-repudiation*. [5]

B. Kriptografi

Menurut Ariyus [1], algoritma kriptografi merupakan langkah- langkah logis bagaimana menyembunyikan pesan dari orang-orang yang tidak berhak atas pesan tersebut dengan melakukan pembangkitan kunci, enkripsi dan dekripsi. Kriptografi bertujuan untuk memberi layanan keamanan (yang juga dinamakan sebagai aspek-aspek keamanan) menurut Schneier (1996), Menezes (1996), dan Wikipedia (2006) yang dikutip oleh Munir [4]

Menurut Munir [4] selain berdasarkan sejarah yang membagi kriptografi menjadi kriptografi klasik dan modern. Maka berdasarkan kunci untuk enkripsi dan deskripsi, kriptografi dapat dibedakan menjadi kriptografi kunci simetri dan kriptografi kunci nirsimetri dan keduanya termasuk dalam kriptografi modern. Sedangkan menurut Ariyus [1] berdasarkan kunci untuk enkripsi dan deskripsi selain kriptografi kunci simetri dan kriptografi kunci nirsimetri ada juga fungsi hash.

C. Algoritma Base64

.Dalam Wahyu, dkk [7], transformasi *Base64* 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 *Base64* ini terdiri dari A..Z, a..z dan 0..9, serta ditambah dengan dua karakter terakhir yang bersimbol yaitu + dan / serta satu buah karakter sama dengan (=) yang digunakan untuk penyesuaian dan menggenapkan data binary atau istilahnya disebut sebagai pengisi pad. Karakter simbol yang akan dihasilkan akan tergantung dari proses algoritma yang berjalan. Dalam *Encoding Base64* dapat dikelompokkan dan dibedakan menjadi beberapa kriteria yang tertera.

Tabel 1 *Encoding Base64* (Josefsson, 2003) dalam Wahyu, dkk [7]

Data 6 bit	Karakter encoding 64	Data 6 bit	Karakter encoding 64	Data 6 bit	Karakter encoding 64	Data 6 bit	Karakter encoding 64
0	A	16	Q	33	h	50	y
1	B	17	R	34	i	51	z
2	C	18	S	35	j	52	0
3	D	19	T	36	k	53	1
4	E	20	U	37	l	54	2
5	F	21	V	38	m	55	3
6	G	22	W	39	n	56	4
7	H	23	X	40	o	57	5
8	I	24	Y	41	p	58	6
9	J	25	Z	42	q	59	7
10	K	26	a	43	r	60	8
11	L	27	b	44	s	61	9
12	M	28	c	45	t	62	+
13	N	29	d	46	u	63	/
14	O	30	e	47	v	pad	=
15	P	31	f	48	w		
16	Q	32	g	49	x		

Menurut Ariyus (2008) Teknik *encoding Base64* sebenarnya sederhana, jika ada satu (*string*) *bytes* yang akan disandikan ke *Base64* maka caranya adalah yang dikutip oleh Wahyu, dkk [7].

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 *decimal*, imana maksimal nilai 6 *bit* dalah 63.
5. Terakhir, jadikan nilai nilai desimal tersebut menjadi indeks untuk memilih karakter penyusun dari *base64* dan maksimal adalah 63 atau indeks ke 64.

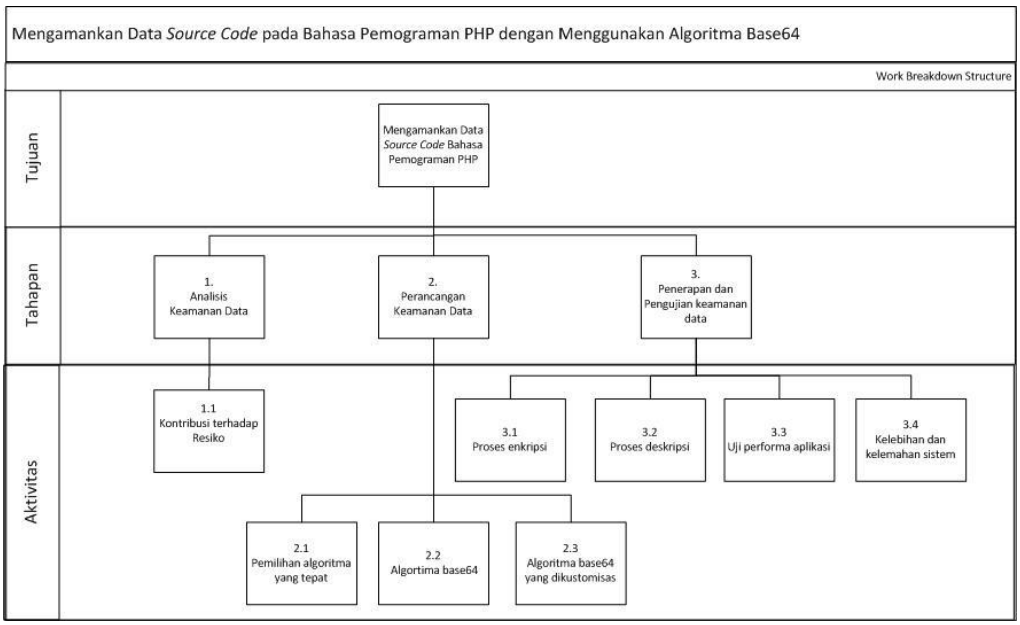
Dan seterusnya sampai akhir *string bytes* yang mau kita konversikan. Jika ternyata dalam proses *encoding* terdapat sisa pembagi, maka tambahkan sebagai penggenap sisa tersebut karakter =. Maka terkadang pada *base64* akan muncul satu atau dua karakter = ().

D. Definisi PHP

PHP atau *Personal Home Page* adalah bahasa pemrograman web atau *scripting language* yang didesain untuk web. PHP dibuat pertama kali oleh satu orang yaitu Rasmus Lerdorf, yang pada awalnya dibuat untuk menghitung jumlah pengunjung pada halaman webnya. Bahasa pemrograman PHP dapat digabungkan dengan HTML dengan terlebih dahulu memberikan tanda tag buka dilanjutkan tanda tanya (**<?**) kemudian ditutup dengan tanda tanya dilanjutkan tanda tag tutup (**?>**). [8]

3. KERANGKA KERJA KONSEPTUAL

Pemodelan data yang digunakan dalam penelitian ini menggunakan *flowchart* seperti yang dijelaskan oleh STTG [6], yang digambarkan dengan pemodelan *work Breakdown Structure* dari Dawson [3], dapat dilihat pada gambar dibawah ini.



Gambar 1 *work breakdown structure*

4. HASIL dan PEMBAHASAN

A. Analisis Keamana Data

Berikut langkah- langkah pengelolaan keamanan sebuah sistem, dilihat dari kontribusi terhadap resiko:

1) Aset (*Asset*)

Aplikasi web sudah dipakai dalam berbagai aspek kegiatan yang menggunakan interaksi antara organisasi dan penggunanya dimana saja dan kapan saja. Tentunya akan banyak data yang bersifat penting dan rahasia dan jika data- data tersebut dilanggar hak aksesnya, bisa berakibat fatal terhadap keberlangsungan sistem itu sendiri. Sebagai contoh:

a. Jika situs resmi internet banking sebuah bank yang melayani nasabahnya diserang dan peretas dapat dengan mudah mendapatkan data pribadi nasabah seperti PIN kartu kredit, mungkin para nasabah akan berpikir ulang mengenai keamanan akan uang yang disimpannya dan bisa para nasabah tersebut beralih ke bank lain yang dirasa lebih aman. Kasus seperti pernah terjadi kepada situs resmi klikbca.com, dimana peretas membuat situs yang mirip dengan situs resmi milik BCA tersebut. Kasus ini berkaitan dengan aspek keamanan kerahasiaan (*privacy/ confidentiality*) dari data- data nasabah dan aspek keamanan otentikasi (*authentication*) dari situs itu sendiri

b. Sebagian besar pendapatan di dunia maya didapat dari iklan dan forum jual beli, seperti situs amazon, kaskus, tokobagus dll. Jika situs- situs tersebut mengalami "*downtime*" selama beberapa jam saja, berapa banyak kerugian dalam hal pendapat yang mereka alami karena *downtime* tersebut, selain itu mungkin setelahnya akan banyak pengguna jasa iklan akan berpindah ke situs lain yang lebih stabil penggunaanya, dengan kata lain situs- situs diatas akan kehilangan pelanggannya. Aspek keamanan yang berhubungan dengan kasus ini, yakni ketersediaan informasi ketika dibutuhkan (*availability*).

c. Fungsi utama dari sebuah situs pastinya untuk menyampaikan sebuah informasi kepada para penggunanya. Jika situs- situs resmi sebuah perusahaan atau pemerintahan diretas dan informasinya dirubah, mungkin pihak- pihak yang membutuhkan informasi tersebut sebagai data bisa melakukan kesalahan dalam melakukan sesuatu yang berasal dari informasi tersebut. Kejadian seperti ini pernah terjadi terhadap situs satreskrim-garut.org, Maret 2013. Dimana halaman awal situs ini dirubah, peretas meninggalkan pesan untuk tidak menggunakan enkripsi gratisan. Kejadian ini akan menurunkan kredibiltas dari kepolisian, dan tentunya mungkin kepercayaan dari masyarakat. Kabar bagusnya peretas hanya

meninggalkan pesan, jika peretas mengubah informasi tertentu tanpa seizin dari yang berhak bisa saja akan memberikan dampak yang lebih besar di masyarakat. Dalam kasus ini, aspek keamanan yang dilanggar yakni integritas data dan *access control*.

Dari contoh di atas, begitu banyak aset yang harus dilindungi sesuai dengan fungsi dari situs itu sendiri, sebut saja data- data pengguna seperti nama, alamat email, PIN kartu kredit, selain itu yang lebih penting bagi perusahaan atau sebuah organisasi yang meluncurkan situs tersebut yakni kepercayaan, keamanan dan kenyamanan para pelanggannya dalam melakukan aktifitasnya yang nantinya akan berdampak terhadap kredibilitas peluncur situs itu sendiri.

2) Ancaman (*threats)*
Serangan- serangan yang terjadi, bisa berasal dari pemakai *(users)* yang teledor seperti memberikan *password* administrator kepada siapa saja atau serangan datang dari pada *crackers* yang dengan sengaja menyerang kelemahan dari sistem untuk mendapatkan apa yang diinginkannya. Oleh sebab itu, harus ada sebuah solusi untuk menekan serangan- serangan dari para *crackers* tersebut.
3) Kelemahan (*Vulnerabilities*)
Bahasa pemograman PHP tidak memerlukan kompilasi dalam penggunaanya, dengan kata lain PHP harus didistribusikan dalam bentuk *source*. Dengan seperti ini, hanya dengan menembus *source* tersebut, para *crackers* dapat mengetahui letak- letak kelemahan dari alur program itu sendiri dan dengan mudah dapat menyerangnya. Selain itu, *source* tersebut dapat dengan mudah disalin, diubah, dihapus sebagian/ seluruhnya tanpa adanya pemberitahuan sebelumnya. Sebagai contoh:

Gambar 2 tampilan awal sebuah *website*

Gambar di atas merupakan tampilan awal dari sebuah *website.* Kemudian seorang oknum pengembang yang melanggar kekayaan intelektual pengembang yang lain dapat merubahnya seperti gambar dibawah ini:

Gambar 3 tampilan awal yang sudah dirubah

Dalam gambar 3 dirubah *header* dan *footer* dari website pada gambar 2, dengan tidak dienkripsinya skrip dari aplikasi web tersebut, oknum pengembang dapat dengan mudah mengakui sebuah karya cipta orang lain. Oleh karena itu, untuk memecahkan masalah tersebut, dalam penelitian ini akan dikembangkan sebuah aplikasi yang dapat mengenkripsi skrip PHP. Adapun spesifikasi dari aplikasi yang akan dikembangkan, adalah:

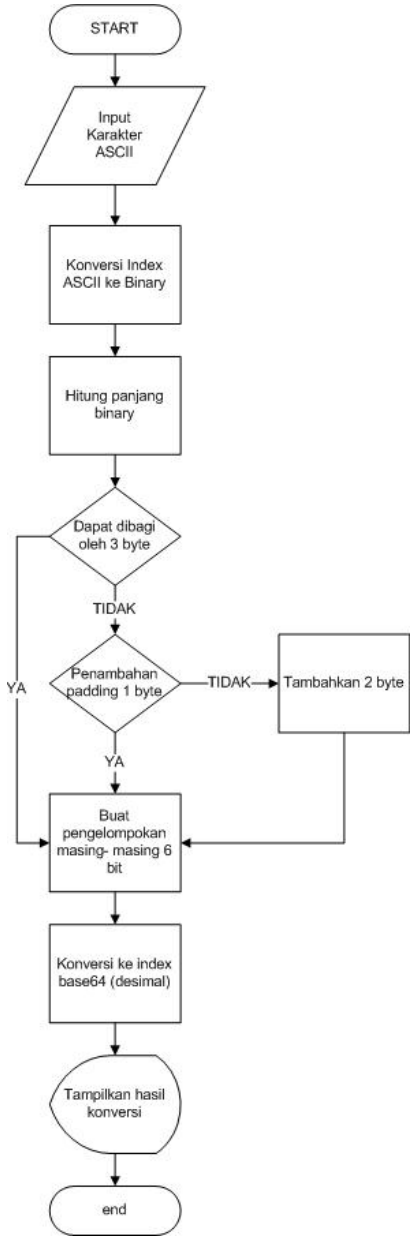
a. *File* berekstensi php yang telah terenkripsi dapat digunakan sebagaimana mestinya di *web browser*, dengan kata lain dalam *file* tersebut tertanam sebuah cara untuk mendeskripsi kode PHP tersebut.
b. Menggunakan algoritma kriptografi yang sesuai dengan kebutuhan nomor satu.
c. Dalam hal merancang aplikasi yang digunakan untuk men-enkripsi *file* berekstensi PHP menggunakan bahasa pemograman C#

B. Perancangan Keamanan Data

1) Pemilihan Kriptografi

Sesuai dengan spesifikasi aplikasi, dimana aplikasi web nantinya akan dihosting berarti harus diperhatikan dalam segi ukuran yang dihasilkan supaya nantinya tidak terlalu memberatkan sistem dalam hal kapasitas programnya. Oleh karena itu dipilihlah kriptografi cipher blok, dikarenakan cipherteks dari cipher blok mempunyai ukuran yang sama dengan plainteksnya. Didalam kriptografi cipher blok, proses enkripsi dan dekripsi dilakukan terhadap sejumlah blok yang terdiri dari sejumlah bit. Panjang bit sudah diketahui sebelumnya dan disesuaikan dengan panjang kunci yang terdiri dari 64 bit atau lebih, inilah yang menyebabkan base64 enkode selalu digunakan dalam kriptografi cipher blok yang secara *default* terdapat dalam fungsi PHP. Oleh karena itu, supaya proses enkripsi dan deskripsi tidak terlalu panjang yang nantinya akan berpengaruh terhadap *respon time* dari aplikasi web yang dienkripsi karena harus secara otomatisasi melakukan deskripsi sendiri, maka penelitian ini hanya menggunakan algoritma base64 untuk proses enkripsi dan deskripsinya.

2) Algoritma Base64



Gambar 4 *flowchart* proses enkripsi

Gambar 5 *flowchart* proses deskripsi

3) Algoritma Base64 Kostumisasi

Kunci dari algoritma base64 terdapat pada kode index-nya, dimana plainteks yang dimasukan akan diproses dan dikonversi kedalam tabel index berdasarkan tabel 1. Oleh karena itu, supaya menghamburkan makna dari hasil enkode yang nantinya akan berpengaruh terhadap hasil dekode para kriptanalis yang berusaha menembus keamanan dari data skrip bahasa pemograman PHP yang telah terenkripsi, dalam penelitian ini susunan dari tabel index base64 akan dirubah, seperti berikut:

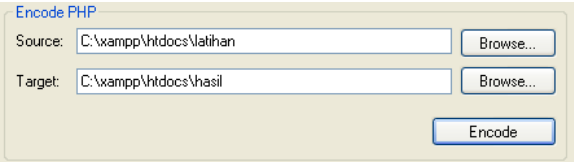
Tabel 2 index base64 kostumisasi

Data 6 bit	**Karakter encoding 64**	**Data 6 bit**	**Karakter encoding 64**	**Data 6 bit**	**Karakter encoding 64**	**Data 6 bit**	**Karakter encoding 64**
0	0	17	f	34	J	51	Z
1	1	18	g	35	K	52	Y
2	2	19	h	36	L	53	X
3	3	20	i	37	M	54	W
4	4	21	j	38	z	55	V
5	5	22	k	39	y	56	U
6	6	23	l	40	x	57	T
7	7	24	m	41	w	58	S
8	8	25	A	42	v	59	R
9	9	26	B	43	u	60	Q
10	+	27	C	44	t	61	P
11	/	28	D	45	s	62	O
12	a	29	E	46	r	63	N
13	b	30	F	47	q	pad	=
14	c	31	G	48	p		
15	d	32	H	49	o		
16	e	33	I	50	n		

C. Penerapan dan Pengujian

1) Proses Enkripsi

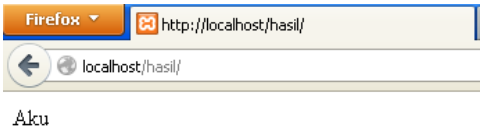
Sebagai contoh terdapat skrip <?php Echo “aku”; ?> dengan nama *file* latihan.php, yang akan dienkripsi dan disimpan ke dalam folder hasil.



Gambar 6 tampilan aplikasi

2) Proses Deskripsi

Proses deskripsi, ditanamkan dalam *file* berekstensi PHP yang telah terenkripsi. Dengan menggunakan fungsi strtr(); index dari fungsi dekode algortima base64 dirubah ke index algoritma base64 yang sudah dikostumisasi, dan fungsi eval(); digunakan untuk mengeksekusi perintah php didalam *file* php. Berikut hasil deskripsi:



Gambar 7 hasil deskripsi di *web browser*

3) Performa Sistem

Uji performa sistem ini yakni dilakukan dengan menghitung rata- rata kecepatan *respon time* dari server php. *Script benchmark* ini menghitung *respon time* yang dibutuhkan php untuk mengeksekusi skrip php yang belum dan sudah terenkripsi. Berikut data *respon time* dari lima kali percobaan yang dilakukan terhadap *file* latihan.php dengan hasil.php:

Tabel 3 *respon time*

Percobaan	Nama File	
	Latihan.php[1]	Hasil.php[1]
1	0.0010089874267578	0.0013260841369629
2	0.0015850067138672	0.0016939640045166
3	0.0010290145874023	0.0023651123046875
4	0.0012528896331787	0.0020802021026611
5	0.0014100074768066	0.0027811527252197

[1]dalam satuan detik

Berdsarkan data di atas, dapat diketahui bahwa respon time PHP untuk mengeksekusi perintah tersebut terjadi fluktuasi namun *respon time* dari *file* hasil.php yang merupakan *file* hasil enkripsi mengalami *respon time* yang lebih lambat dari *respon time* latihan.php.

4) Kelebihan dan Kelemahan Sistem

a. Kelebihan Sistem
   1. Index atau kunci dari algoritma base64 telah dirubah, dengan tujuan menghamburkan makna dari plainteks ketika peretas melakukan kriptanalisis.
   2. Kapasitas *file* dari yang belum dan sudah dienkripsi relatif sama karena jumlah bit *file* yang sudah dan belum dienkripsi sama.
   3. Dapat melakukan proses enkripsi pada *file* php yang isinya bercampur dengan html, javascript dan css dan hanya melakukan enkripsi terhadap skrip php.

b. Kelemahan Sistem
   1. Kecepatan *respon time script* php lebih lambat jika dibandingkan dengan *respon time script* sebelum enkripsi.
   2. Penggunaan *memory* dalam mengeksekusi *file* php yang terenkripsi bertambah, karena proses dekode membutuhkan buffer.

5. KESIMPULAN

Kesimpulan yang dapat diperoleh dari hasil yang telah dikembangkan mengenai cara mengamankan skrip bahasa pemograman PHP, adalah sebagai berikut:

1. Dengan adanya cara pengamanan ini, pengembang aplikasi yang menggunakan bahasa pemograman PHP dapat menyembunyikan skrip php supaya tidak mudah disalin, diubah sebagian/ seluruhnya oleh orang yang tidak berhak.
2. Integritas dari aplikasi yang telah dienkripsi akan lebih terjaga, karena skrip yang sudah dienkripsi tidak dapat diubah.
3. Kelemahan- kelemahan dari alur program yang terdapat dalam aplikasi php dapat terjaga secara otomatis, karena skrip aplikasi php tidak bisa dibaca kembali, kecuali menggunakan skrip yang belum dienkripsi.

DAFTAR PUSTAKA

[1] Ariyus, D. (2006). *KRIPTOGRAFI Keamanan Data dan Komunikasi*. Yogyakarta: Graha Ilmu.

[2] Builtwith. *PHP usage Statistics*, diakses 13 Maret 2013, dari Google.com World Wide Web: http://trends.builtwith.com/framework/PHP

[3] Dawson, C.W. (2005). *Project on computing and information system : a student's guide*. England : Pearson Education Limited.

[4] Munir, R. (2006) Kriptografi. Bandung: Informatika

[5] Rahardjo, B. (2005). Keamanan Sistem Informasi Berbasis Internet. Jakarta: PT INDOCISC.

[6] STTG. (2009) Modul Praktikum Algoritma dan Pemograman dalam Bahasa Pascal dan C. Garut: Sekolah Tinggi Teknologi Garut

[7] Wahyu C, F. & P Rahangiar, A. & de Fretes, F. (2012). *Penerapan Algoritma Gabungan Rc4 dan Base64 pada Sistem Keamanan E-Commerce, Seminar Nasional Aplikasi Teknologi Informasi 2012*, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana. Yogyakarta: Universitas Kristen Satya Wacana dari Google.com World Wide Web: http://journal.uii.ac.id/index.php/Snati/article/viewFile/2873/2628

[8] Wikipedia. *Pengertian dan sejarah PHP*, diakses 25 Desember 2012, dari Google.com World Wide Web: http://id.wikipedia.org/wiki/php