RANCANG BANGUN PHP 5 ENCODER

Ali Latiful Aprianto[1], Idris Winarno, SST M.Kom[2]
[1]Mahasiswa Jurusan Teknik Informatika, [2]Dosen Jurusan Teknik Informatika
Jurusan Teknik Informatika Politeknik Elektronika Negeri Surabaya Institut Teknologi Sepuluh Nopember
Kampus ITS Sukolilo, Surabaya 60111, Indonesia
Tel: +62 (31) 594 7280; Fax: +62 (31) 594 6114

ABSTRAK

Hanya dalam beberapa tahun PHP telah cepat berevolusi dari bahasa kecil menjadi sebuah bahasa pengembangan web yang kuat. Sekarang PHP digunakan pada lebih dari 14 juta situs web dan saat ini PHP semakin stabil dan lebih diperluas dari versi-versi pendahulunya. Sayangnya aplikasi hasil PHP harus didistribusikan dalam bentuk source code, sehingga memiliki beberapa kekurangan dan celah keamanan. Beberapa kekurangan tersebut salah satunya adalah source code dapat dengan mudah disalin, diubah, ataupun digunakan sebagian dalam aplikasi lainnya tanpa ada pemberitahuan. Selain itu source code yang tidak terenkripsi membuat aplikasi yang dibangun sangat rentan, karena source code dapat mengungkapkan beberapa kelemahan dari aplikasi yang dibuat.

Karena itulah diperlukan adanya suatu solusi yang dapat menyembunyikan source code aplikasi PHP yang akan didistribusikan, diantaranya dengan menggunakan PHP Encoder. Dimana PHP Encoder dalam tugas akhir ini menggunakan metode enkripsi blowfish untuk meng-encode atau mengenkripsi data yaitu merubah data dari bentuk plain-text menjadi chipper-text untuk menyembunyikan source code dari aplikasi php.

PHP encoder ini dapat semakin dikembangkan dengan cara menambahkan fitur cache. Sehingga sistem ini dapat melakukan proses eksekusi file php terenkripsi dengan respon time yang tidak kalah dengan file php tanpa enkripsi.

I. PENDAHULUAN

1.1 LATAR BELAKANG MASALAH

Pada saat ini PHP digunakan pada lebih dari 14 juta situs Web dan saat ini PHP semakin stabil dan lebih diperluas dari versi-versi pendahulunya. Sayangnya aplikasi hasil PHP harus didistribusikan dalam bentuk source, sehingga memiliki beberapa kekurangan dan celah keamanan. Beberapa kekurangan tersebut salah satunya adalah source code dapat dengan mudah disalin, diubah, ataupun digunakan sebagian dalam aplikasi lainnya tanpa ada pemberitahuan. Selain itu source code yang tidak terenkripsi membuat aplikasi yang dibangun sangat rentan, karena source code dapat mengungkapkan beberapa kelemahan dari aplikasi yang dibuat.

Karena itulah diperlukan adanya suatu solusi yang dapat menyembunyikan source code aplikasi PHP yang akan didistribusikan, diantaranya dengan menggunakan PHP Encoders. Dimana PHP Encoder dapat digunakan untuk meng-encode atau mengenkripsi data yaitu merubah data dari bentuk plain-text menjadi chipper-text dengan metode tertentu untuk menyembunyikan source code dari aplikasi php. Selain menyembunyikan source code PHP Encoder seringkali juga berfungsi untuk melakukan identifikasi mesin, apakah aplikasi php berjalan pada mesin yang telah ditentukan dengan melihat IP address, domain name, ataupun MAC address dari suatu web server sebagai penanda. PHP Encoder juga bisa difungsikan untuk menginputkan suatu masa trial pada source php, sehingga developer dapat menambahkan masa trial pada aplikasi php yang telah dibangun tanpa harus khawatir baris kode dari masa trial diubah oleh user atau client.

1.2 TUJUAN

Rancang bangun PHP5 Encoder ini pada dasarnya memiliki beberapa tujuan, antara lain:

1. Mengamankan kelemahan yang terdapat dalam source PHP
2. Menjamin kekayaan intelektual seorang programmer aplikasi PHP
3. Memastikan integritas aplikasi

1.3 RUMUSAN MASALAH

Beberapa masalah yang tercakup dalam pembuatan proyek akhir in antara lain adalah :

1. Cara membuat loader yang digunakan untuk membaca file php yang telah terenkripsi.
2. Pemilihan metode yang paling optimum untuk digunakan sebagai metode enkripsi pada php5.
3. Cara membuat ekstensi encoder-loader yang compatible dengan php5.

1.4 BATASAN MASALAH

Dalam pengerjaan proyek akhir ini terdapat beberapa batasan masalah antara lain :

1. Dalam rancang bangun PHP Encoder ini tidak termasuk implementasi cache.
2. File yang dapat dienkripsi oleh PHP Encoder ini hanya file dengan ekstensi php.
3. Uji coba dilakukan pada OS Linux (Ubuntu) dengan menggunakan PHP versi 5.2.6.

II. TEORI PENUNJANG

PHP encoders adalah program yang dirancang untuk melindungi hak kekayaan intelektual anda dengan meng-encode atau meng-obfuscating kode sumber aplikasi php, selain itu dapat pula secara opsional memberikan lisensi, membatasi instalasi dengan melihat IP/domain, dan menyediakan fitur batas waktu (berguna untuk versi trial).

PHP Encoders diperlukan karena mendistribusikan aplikasi php yang tidak terenkripsi sangat tidak aman. Source code dapat dengan mudah disalin, diubah, ataupun digunakan sebagian dalam aplikasi lainnya tanpa ada pemberitahuan. Selain itu source code yang tidak terenkripsi membuat aplikasi yang dibangun sangat rentan, karena source code dapat mengungkapkan beberapa kelemahan dari aplikasi yang dibuat.

Beberapa solusi yang ada dalam pasaran dapat dibagi menjadi 2 bagian, yaitu:

- **Source Code Obfuscators**

Source code dari aplikasi dibuat untuk sulit dipahami dan dilakukan perubahan. Contoh obfuscator adalah dengan menghapus jeda baris, spasi, comment, nama variabel, nama fungsi, dll. Mengembalikan source code yang telah diobfuscator sangatlah mudah, solusi ini sangatlah tidak aman, selain itu dapat menurunkan kinerja eksekusi php dan tidak 100% kompatibel dengan kode php. Solusi ini cocok untuk beberapa orang karena tergolong murah bahkan gratis.

- **Encoders**

Menyembunyikan (beberapa juga melakukan optimize) source code PHP, mencompilasi source code menjadi bytecode dan menghilangkan source code asli. Solusi ini sangat stabil dan sangat sulit untuk dikembalikan ke source code awal tanpa pengetahuan tertentu. Encoder terkadang dapat mengelola lisensi dan membuat suatu file dapat berakhir pada masa waktu tertentu (periode percobaan).

Tipe ini dibagi menjadi 2 bagian utama:

- Encoder: program yang meng-encode atau mengenkripsi source code php
- Loader (decoder): program yang didesain khusus untuk mendecode source code yang telah terenkripsi dan menjalankannya. Loader ini ditempatkan dalam server untuk dapat menjalankan aplikasi yang telah ter-encode.

III. PERANCANGAN SISTEM

3.1 METODE YANG DIGUNAKAN

Beberapa metode atau tahapan yang digunakan pada pengerjaan proyek akhir ini antara lain:

1. Metode Enkripsi Blowfish

Metode enkripsi yang digunakan untuk melakukan proses encode-decode adalah tipe enkripsi simetrik dengan menggunakan metode blowfish. Metode ini dipilih karena kemudahan implementasi dan proses enkripsi-dekripsi yang cepat.

Blowfish merupakan blok cipher 64-bit dengan panjang kunci variabel. Algoritma ini terdiri dari dua bagian: key expansion dan enkripsi data. Key expansion merubah kunci yang dapat mencapai 448 bit menjadi beberapa array subkunci (subkey) dengan total 4168 byte. Enkripsi data terdiri dari iterasi fungsi sederhana sebanyak 16 kali. Setiap putaran terdiri dari permutasi kunci-dependent dan substitusi kunci- dan data-dependent. Semua operasi adalah penambahan dan XOR pada variable 32-bit. Tambahan operasi lainnya hanyalah empat penelusuran tabel (table lookup) array berindeks untuk setiap putaran.

2. Proses Encode

Proses encode pada proyek akhir ini adalah suatu proses yang digunakan untuk melakukan encoding pada script php yang ingin disembunyikan source codenya. Proses ini memerlukan inputan berupa script yang ada di dalam file php. Biasanya script ini berada diantara tag <?php … ?>. Proses ini akan menghasilkan output sebuah string hasil encode yang telah diubah kedalam bentuk hexadecimal.

3. Proses Encrypt

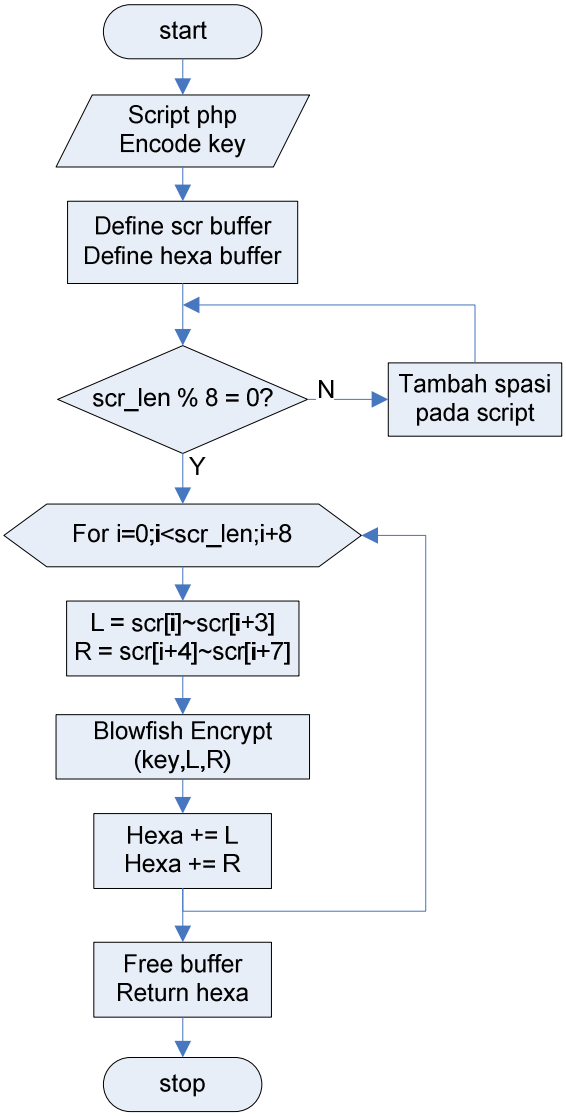
Berbeda dengan proses encode yang membutuhkan input script, pada proses encrypt yang digunakan sebagai input adalah nama file dari file php yang ingin disembunyikan source codenya. Proses encrypt ini menghasilkan output sebuah file yang telah diset nama filenya pada saat melakukan proses encrypt. File hasil output ini berisi sebuah fungsi yang memanggil proses decode disertai dengan parameter berupa karakter hexadecimal hasil proses encrypt.

4. Proses Decode

Proses decode ini merupakan proses pengembalian source yang telah diencode. Proses ini terjadi ketika sebuah web browser (client) membuka halaman web php yang didalamnya terdapat fungsi decode. Fungsi ini mengubah parameter inputan dari hexadecimal menjadi string kemudian melakukan proses decrypt dengan algoritma blowfish sesuai dengan key yang ada dalam loader yang telah decompile. Fungsi ini tidak memberikan return value apapun, karena script php yang berhasil didecode akan langsung di eksekusi di dalam loader tersebut, tanpa menyimpannya ke dalam file temporary.

3.2 FLOWCHART PROSES ENCODE

Proses encode dimulai dengan user memasukkan script php dan encode key. Kemudian script serta key tersebut akan dikirim sebagai parameter fungsi pada php extension. Berikut ini flowchart dari proses encode yang terjadi di dalam php extension.

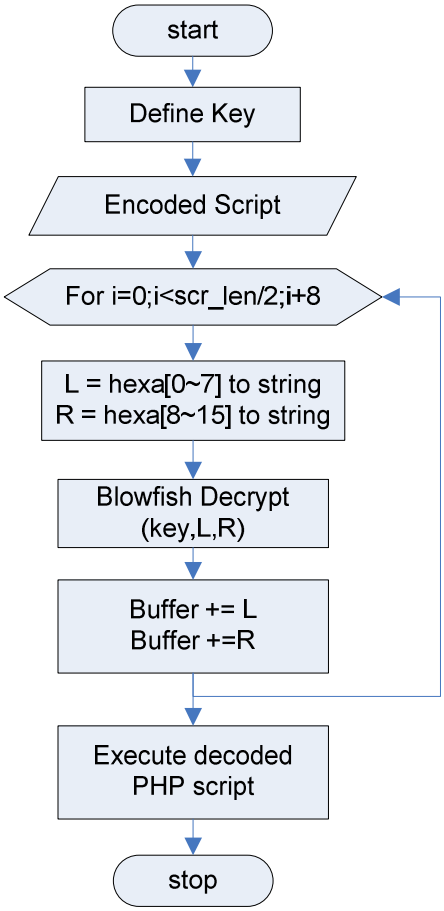


Gambar 3.10 Flowchart Encode dalam PHP Extension

Dari flowchart 3.10 tampak bahwasanya fungsi encode yang terjadi di dalam php extension membutuhkan 2 input parameter, yaitu script php dan key. Script php yang masuk pada proses ini akan dipaksakan untuk mempunyai panjang kelipatan 8, dengan cara menambahkan karakter spasi di akhir script. Hal ini dilakukan karena proses encrypt blowfish yang melakukan enkripsi tiap 8 karakter, seperti tampak dari loop yang dilakukan. Dalam looping tiap 8 karakter ini masih dipecah lagi menjadi 2 bagian sama besar yaitu L dan R yang kemudian dikirim ke fungsi encrypt untuk dilakukan enkripsi menggunakan algoritma blowfish. String yang telah terenkripsi kemudian dikonversi menjadi bentuk hexadecimal. Setelah looping selesai script php yang telah terencode dan telah dalam bentuk hexadecimal akan dikembalikan pada proses yang memanggilnya.

3.3 FLOWCHART PROSES DECODE

Proses decode adalah proses yang dimulai ketika ada permintaan dari sebuah web browser untuk membuka file php yang terenkripsi. Proses ini berjalan di dalam sebuah extension di dalam php. Berikut ini flowchart dari proses decode tersebut.



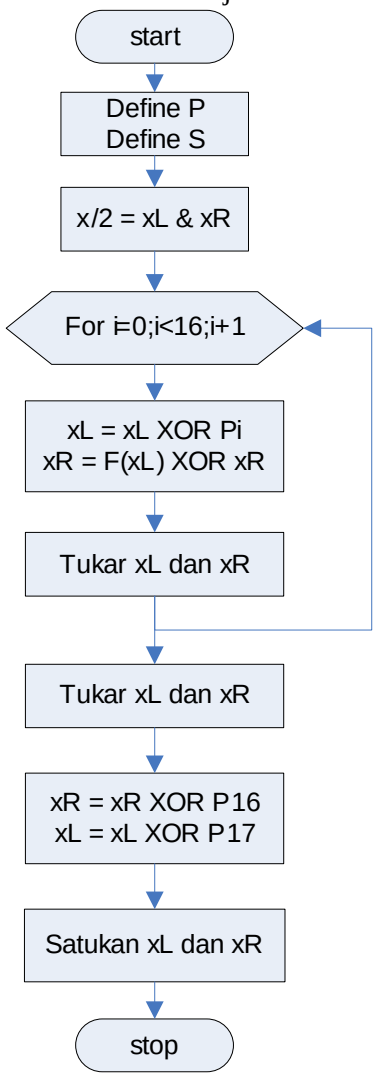
Gambar 3.13 Flowchart Proses Decode pada PHP Extension

Gambar 3.13 menunjukkan bahwa proses decode diawali dengan mengambil parameter script yang terenkripsi. Script ini kemudian di baca setiap 16 karakter, dimana masih dibagi lagi menjadi 2 bagian yaitu 8 karakter. 8 karakter ini kemudian dikonversi dari hexadecimal ke dalam bentuk string. Tiap 8 karakter hexadecimal akan membentuk 4 karakter string. Karakter ini kemudian dimasukkan ke dalam algoritma blowfish untuk dilakukan proses decode. Hasil proses decode akan disatukan ke dalam sebuah buffer yang telah disiapkan. Setelah semua string terdecode maka script php ini akan langsung dieksekusi saat itu juga, tanpa menyimpan hasil decode dalam sebuah file temporary. Sehingga script asli tidak akan terlihat.

3.4 FLOWCHART ALGORITMA BLOWFISH

Blowfish menggunakan subkunci yang besar. Kunci tersebut harus dihitung sebelum enkripsi atau dekripsi data. Namun dalam tugas akhir ini subkunci tersebut ditulis secara hardcoded di dalam source code blowfish. Blowfish adalah algoritma yang menerapkan jaringan Feistel (*Feistel Network*) yang

terdiri dari 16 putaran. Untuk alur algoritma enkripsi dengan metoda Blowfish dijelaskan sebagai berikut



Gambar 3.14 Flowchart Algoritma Blowfish

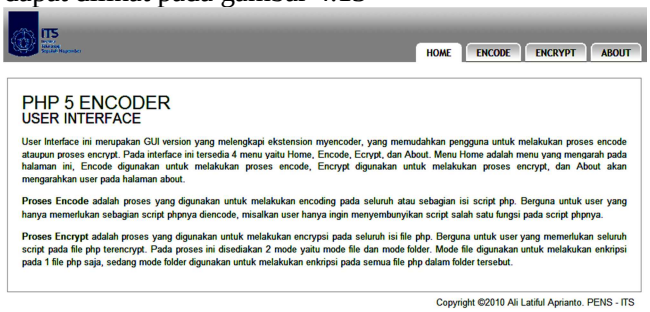
Flowchart algoritma blowfish pada gambar 3.14 dapat dijelaskan sebagai berikut:

1. Bentuk inisial P-array sebanyak 18 buah (P1,P2,.....P18) masing-masing bernilai 32-bit.
2. Bentuk S-box sebanyak 4 buah masing-masing bernilai 32-bit yang memiliki masukan 256.
3. Plaintext yang akan dienkripsi diasumsikan sebagai masukan, Plaintext tersebut diambil sebanyak 64-bit, dan apabila kurang dari 64-bit maka kita tambahkan bitnya, supaya dalam operasi nanti sesuai dengan datanya.
4. Hasil pengambilan tadi dibagi 2, 32-bit pertama disebut XL, 32-bit yang kedua disebut XR
5. Selanjutnya lakukan operasi XL = XL xor Pi dan XR = F(XL) xor XR
6. Hasil dari operrasi diatas ditukar XL menjadi XR dan XR menjadi XL.
7. Lakukan sebanyak 16 kali, perulangan yang ke-16 lakukan lagi proses penukaran XL dan XR.
8. Pada proses ke-17 lakukan operasi untuk XR = XR xor P16 dan XL = XL xor P17.
9. Proses terakhir satukan kembali XL dan XR sehingga menjadi 64-bit kembali.

IV. PENGUJIAN DAN ANALISIS

4.1 ENCODE DAN ENCRYPT

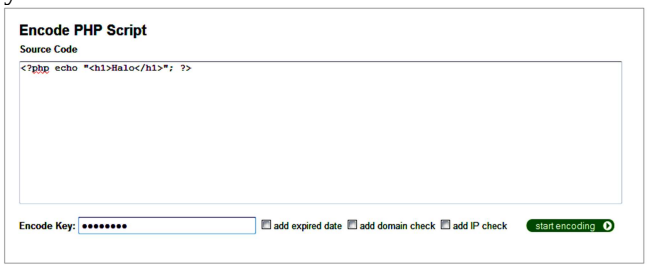
Sebelum dapat menggunakan GUI untuk melakukan encrypt, anda harus mengcopy folder myencoder ke dalam folder /var/www. Folder myencoder ini dapat ditemukan dalam folder GUI pada cd yang ada bersama dengan buku TA ini. Setelah tercopy maka anda dapat menggunakan GUI untuk proses Encode. Untuk memanggil interface ini cukup dengan membuka alamat http://localhost/myencoder. Tampilan dari GUI ini dapat dilihat pada gambar 4.13



Gambar 4.13 GUI dari PHP 5 Encoder

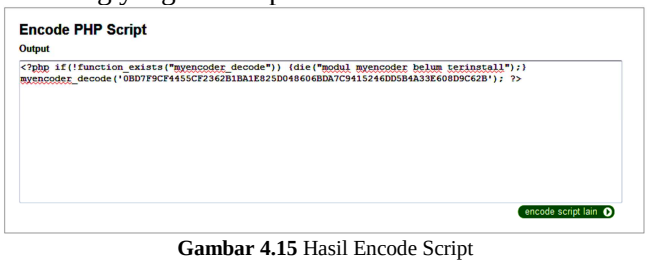
4.1.1 MELAKUKAN ENCODE SCRIPT

Untuk melakukan encode dapat dengan memilih encode yang ada pada menu. Dalam percobaan ini script php yang akan diencode adalah script <?php echo "<h1>Halo</h1>"; ?>. Key yang digunakan dalam percobaan ini adalah key yang sesuai dengan key yang ada pada source myencoder.c, yaitu “ali-demo”.



Gambar 4.14 Percobaan Encode Script

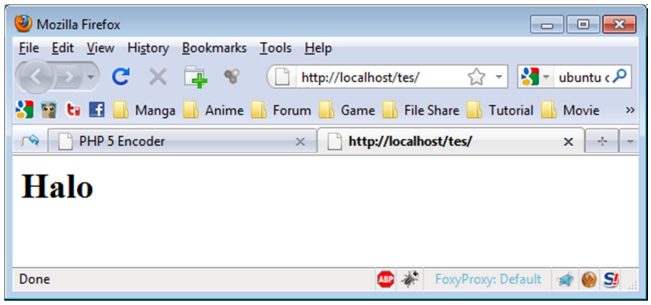
Setelah memasukkan script dan encode key seperti tampak pada gambar 4.14 maka kita dapat langsung melakukan encoding dengan menekan tombol start encoding yang berada pada lokasi kanan bawah.



Gambar 4.15 Hasil Encode Script

Gambar 4.15 merupakan gambar dari halaman hasil proses encoding. Dalam halaman ini terlihat adanya script php yang telah terencode. Dimana di dalam script php ini memanggil fungsi myencoder_decode dengan parameter script php yang terencode dan

dalam bentuk hexadecimal. Jika script tersebut dijalankan maka hasilnya akan tampak seperti gambar 4.16.

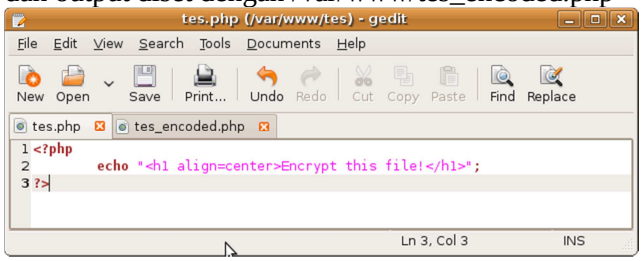


Gambar 4.16 Hasil Running Script Yang Terencode

Pada gambar ini terlihat adanya kata “Halo”, hal ini sesuai dengan code script php yang dilakukan proses encode yaitu <?php echo "<h1>Halo</h1>"; ?>. Dari hasil ini dapat disimpulkan bahwa proses encoding berhasil.

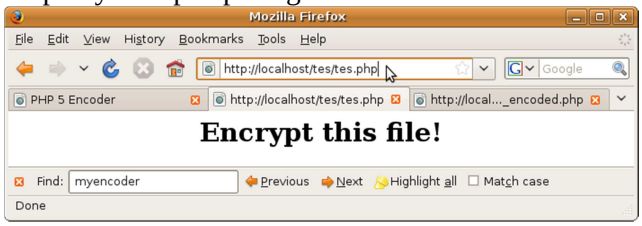
4.1.2 MELAKUKAN ECRYPT FILE

Untuk melakukan encrypt dapat dengan memilih encrypt pada pilihan menu. Percobaan proses encrypt file ini akan dilakukan pada file /var/www/tes/tes.php dan output diset dengan /var/www/tes_encoded.php



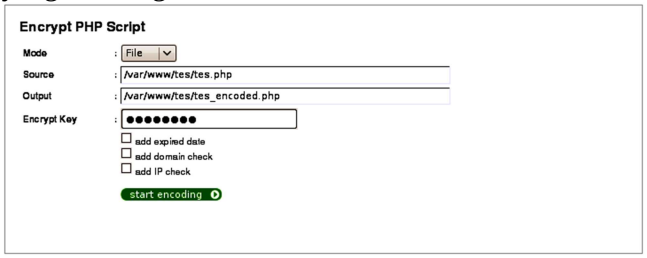
Gambar 4.17 Isi dari File yang Akan Diencrypt

Gambar 4.17 merupakan tampilan isi dari file /var/www/tes/tes.php yang akan diencrypt. Jika script tersebut dipanggil dalam web browser maka hasil outputnya tampak pada gambar 4.18



Gambar 4.18 Hasil Output File Sebelum Encrypt

Untuk melakukan enkripsi pada 1 file maka mode yang akan digunakan adalah mode file.

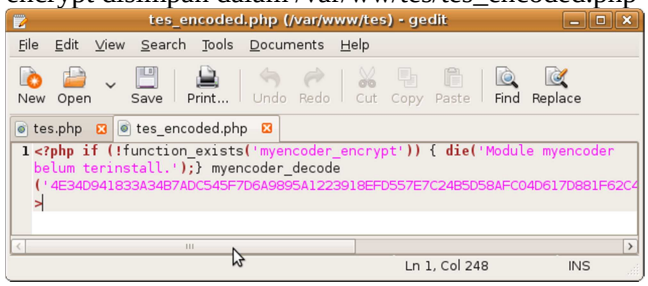


Gambar 4.19 Encrypt File PHP

Setelah form telah terisi seperti pada gambar 4.19 maka kita dapat melakukan proses enkripsi file dengan menekan tombol start encoding.

Gambar 4.20 Halaman Status Encrypt

Gambar 4.20 menunjukkan bahwa file yang kita inputkan telah sukses diencrypt, dengan output file encrypt disimpan dalam /var/ww/tes/tes_encoded.php



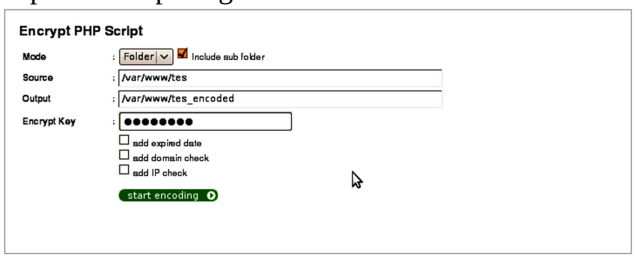
Gambar 4.21 Isi Dari File yang Dienkripsi

Gambar 4.21 Merupakan isi dari file output hasil proses enkripsi file. Dalam file ini dapat diketahui bahwa source code sebelum enkripsi telah berhasil disembunyikan. Apabila file ini dipanggil dalam web browser maka outputnya akan tampak seperti pada gambar 4.22

Gambar 4.22 Hasil Output File Setelah Diencrypt

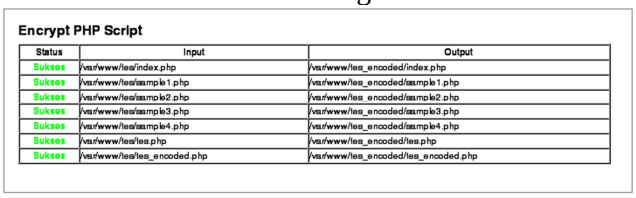
Dari hasil gambar 4.22 dan jika dibandingkan dengan gambar 4.18 maka dapat disimpulkan bahwa proses enkripsi file telah berhasil dengan baik, karena hasil eksekusi file sebelum dan sesudah proses enkripsi adalah sama.

Proses enkripsi pada GUI dapat pula menggunakan mode folder, mode ini digunakan untuk mempermudah user dalam melakukan proses enkripsi pada banyak file php. Selain itu dengan menggunakan mode folder struktur file yang ada dalam folder input dan output adalah sama, sehingga hasil enkripsi dapat langsung didistribusikan. Percobaan penggunaannya dapat dilihat pada gambar 4.23.



Gambar 4.23 Proses Enkripsi dengan Mode Folder

Setelah semua input telah diset seperti pada gambar 4.23, maka proses enkripsi dapat dimulai dengan menekan tombol start encoding.



Encrypt PHP Script

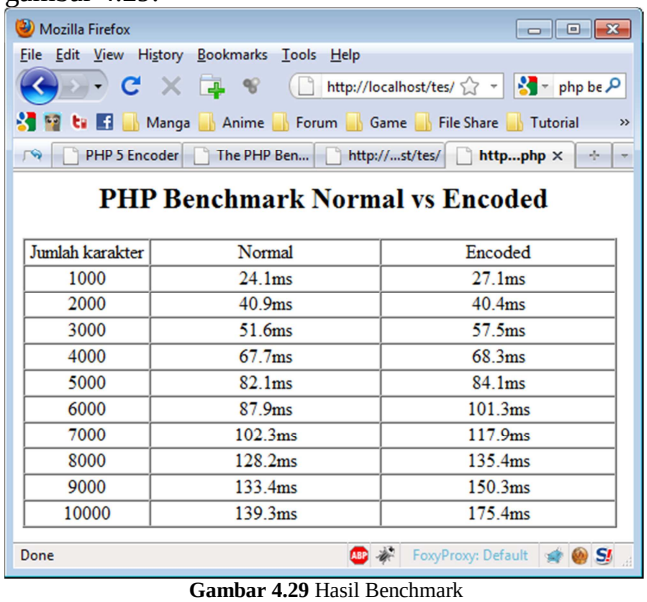
Status	Input	Output
Sukses	/var/www/tes/index.php	/var/www/tes_encoded/index.php
Sukses	/var/www/tes/sample1.php	/var/www/tes_encoded/sample1.php
Sukses	/var/www/tes/sample2.php	/var/www/tes_encoded/sample2.php
Sukses	/var/www/tes/sample3.php	/var/www/tes_encoded/sample3.php
Sukses	/var/www/tes/sample4.php	/var/www/tes_encoded/sample4.php
Sukses	/var/www/tes/tes.php	/var/www/tes_encoded/tes.php
Sukses	/var/www/tes/tes_encoded.php	/var/www/tes_encoded/tes_encoded.php

Gambar 4.24 Status Enkripsi dengan Mode Folder

Gambar 4.24 menunjukkan bahwa semua file yang ada di dalam folder /var/www/tes telah berhasil dienkripsi.

4.2 UJI PERFORMA SISTEM

Uji performa sistem ini dilakukan dengan menghitung rata-rata kecepatan respon time dari server php. Script benchmark ini menghitung respon time yang dibutuhkan php untuk mengeksekusi fungsi strrev() sebanyak 10000 kali untuk tiap-tiap jumlah karakter yang berbeda. Source code yang digunakan untuk proses benchmark dapat dilihat pada lampiran. Hasil perbandingan benchmark dapat dilihat pada gambar 4.29.



Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/tes/

Manga Anime Forum Game File Share Tutorial

PHP 5 Encoder | The PHP Ben... | http://...st/tes/ | http...php

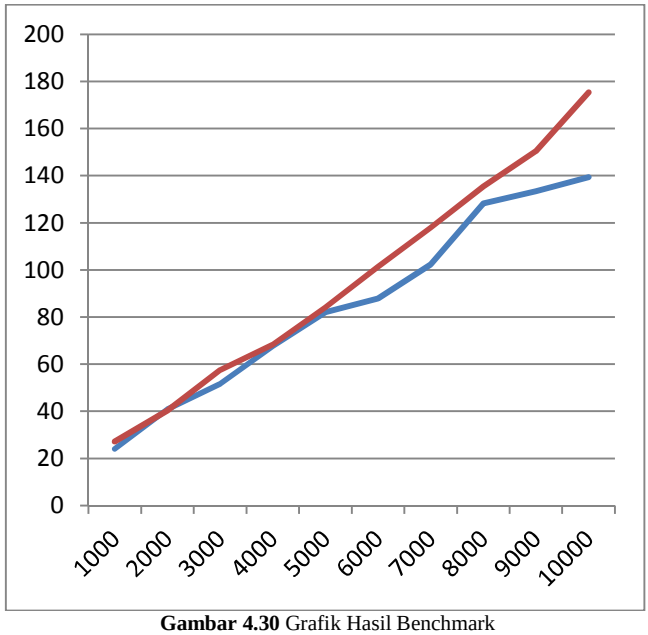
PHP Benchmark Normal vs Encoded

Jumlah karakter	Normal	Encoded
1000	24.1ms	27.1ms
2000	40.9ms	40.4ms
3000	51.6ms	57.5ms
4000	67.7ms	68.3ms
5000	82.1ms	84.1ms
6000	87.9ms	101.3ms
7000	102.3ms	117.9ms
8000	128.2ms	135.4ms
9000	133.4ms	150.3ms
10000	139.3ms	175.4ms

Done FoxyProxy: Default

Gambar 4.29 Hasil Benchmark

Dari gambar 4.29 dapat dibuat sebuah grafik perbandingan antara kecepatan eksekusi normal dengan kecepatan eksekusi file yang terencode. Gambar grafik tersebut dapat dilihat pada gambar 4.30



Gambar 4.30 Grafik Hasil Benchmark

Dari grafik pada gambar 4.30 ini dapat disimpulkan bahwa respon time yang digunakan oleh script yang ternkripsi semakin jauh jaraknya dari respon time eksekusi normal ketika jumlah karakter yang perlu untuk didecode semakin bertambah. Dari sini dapat disimpulkan bahwa menggunakan enkripsi sedikit memperlambat waktu respon eksekusi script php.

4.3 KELEBIHAN DAN KELEMAHAN SISTEM

Dalam suatu sistem pasti terdapat kelebihan maupun kekurangannya. Berikut ini akan disebutkan beberapa kelebihan dan kekurangan dari sistem php encoder ini.

4.3.1 KELEBIHAN SISTEM

Kelebihan yang ada dalam php encoder ini adalah sebagai berikut:

1. Mampu melakukan proses encode dengan ukuran file sampai 1,4MB, dimana ukuran ini termasuk ukuran yang fantastis untuk sebuah file php.
2. Dapat melakukan proses enkripsi pada sebagian isi file php atau pada fungsi-fungsi yang dianggap penting.
3. Dapat melakukan proses enkripsi pada file php yang isinya bercampur dengan html, javascript dan css.
4. Key yang digunakan dalam proses encode maupun decode dapat diset sesuai dengan keinginan.

4.3.2 KELEMAHAN SISTEM

Kelemahan-kelemahan yang ada dalam sistem ini yaitu:

1. Memerlukan proses compile dan implementasi yang cukup sulit bagi para pemula.
2. File yang dienkripsi akan memiliki ukuran file 2 kali lipat dari ukuran file semula. Hal ini dikarenakan file hasil enkripsi ditulis dalam bentuk hexadecimal, dimana 1 karakter biasa membutuhkan 2 karakter hexadecimal.

3. Penggunaan memory dalam mengeksekusi file php yang terenkripsi bertambah. Dikarenakan proses encode maupun decode membutuhkan buffer.
4. Kecepatan respon time script php semakin pelan jika dibandingkan dengan respon time script sebelum enkripsi.

V. PENUTUP

5.1 SIMPULAN

Berdasarkan analisa dari beberapa pengujian yang dilakukan pada bab sebelumnya, kesimpulan yang didapatkan adalah:

1. Dengan menggunakan php encoder ini seorang pengembang aplikasi php dapat menyembunyikan source code php yang tidak ingin diberikan.
2. Integritas aplikasi yang dienkripsi akan lebih terjaga, karena source code php yang terenkripsi tidak dapat diubah.
3. Kelemahan-kelemahan sistem yang ada pada source code php secara otomatis terjaga, karena source code tidak bisa dibaca tanpa dilakukan proses decrypt terlebih dahulu.

1.2 SARAN

Hal yang perlu diperhatikan untuk mengembangkan sistem ini lebih lanjut yaitu perlunya mengatasi kelemahan-kelemahan sistem yang telah dituliskan pada bab sebelumnya, serta menambahkan fitur cache. Sehingga php encoder ini dapat melakukan proses eksekusi file php terenkripsi dengan respon time yang tidak kalah dengan file php tanpa enkripsi.

DAFTAR PUSTAKA

[1] Golemon, Sara (2006). *Extending and Embedding PHP*. United States of America: Sams.

[2] Schlossnagle, George (2004). *Advanced PHP Programming.* United States of America: Sams.

[3] *Pengenalan PHP*, from http://www.deptan.go.id/pusdatin /admin/RB/Programming/Materi%20PHP.pdf, 18 Juli 2010

[4] Suryana Aulya, *Enkripsi*, from http://www.ajaib.us/dl/ Kriptografi.pdf, 18 Juli 2010

[5] Syafari, Anjar, *Sekilas Tentang Enkripsi Blowfish.* IlmuKomputer.com