DESAIN WEB SECURE LOGIN DENGAN ALGORITMA ENKRIPSI SIMETRI RC-6

Arkhan Subari[1)], Mustafid[2)], Kodrat Iman Satoto[3)]
Magister Sistem Informasi Program Pascasarjana Universitas Diponegoro

Abstract

Authentication techniques that use at many web pages and easy to do is use user-id and password. However, these techniques are vulnerable to theft user-id and password when sent from client to server. For that given an alternative security by encrypting the user-id and password at client side before being sent to the server. The algorithm used is symmetric algorithm RC-6, designed with javascript on the client side and PHP on the server side.

Based on RC-6 Symmetric encryption algorithm, the research done by creating a generating keys script for encryption and decryption, encryption RC-6 with javascript, decryption RC-6 with PHP and the design of a prototype web page with a login that already uses encryption.

Using the program fidller and wireshark shows that a web page with login form that does not use encryption to send user-id and password in plaintext form so easily obtained by the sniffer. While in the web pages that use encryption, user-id and password is sent in the form of ciphertext. The addition of a web page access time is shown by firebug, where on the web pages that use encryption are adding an average access time of 64.67 ms.

Keywords : web, login, encryption, decryption, RC-6, PHP, javascript, fidller, wireshark, firebug

PENDAHULUAN

Latar Belakang

Dalam teknologi web, autentifikasi digunakan sebagai sarana untuk mengakses halaman web yang bersifat rahasia dan terbatas. Salah satu metode yang paling banyak digunakan adalah dengan memakai *user-id* dan password yang dimasukkan pada form login. Selain murah dan tidak memerlukan perangkat tambahan, penggunaan *user-id* dan password juga nyaman. *User* hanya perlu menghapal *user-id* dan password kemudian dapat melakukan login dimanapun (Yang, 2009).

Namun demikian penggunaan *user-id* dan password bukannya tanpa kelemahan. User sering kali memilih *user-id* dan password yang pendek dan lemah sehingga mudah dicuri dengan teknik *brute force* (Yang, 2009). Selain itu format standar dari form login akan mengirimkan *user-id* dan password dari client ke server dalam format *plaintext* atau teks asli. Dalam format ini, sangat mudah bagi para hacker untuk mendapatkan data *user-id* dan password yang valid dan dapat digunakan pada form login yang dimaksud (Chakrabarti dan Singhal, 2007).

Untuk menjaga agar *user-id* dan password tidak mudah dibaca oleh hacker diperlukan proses pengamanan data *user-id* dan password tersebut. Alternatif proses pengamanan yang ditawarkan adalah dengan melakukan enkripsi di sisi client sebelum data dikirimkan ke server melalui internet. Dengan demikian yang dikirimkan melalui jaringan internet adalah *ciphertext*. Format *chipertext* juga dapat melindungi *user-id* dan password dari pencurian dengan teknik *brute force* (Halevi dan Krawczyk, 1998). Selanjutnya pada sisi server dilakukan dekripsi kembali data sehingga didapatkan data asli.

Salah satu algoritma enkripsi adalah algoritma RC-6. RC-6 merupakan algoritma cipher blok yang didaftarkan ke NIST yang diajukan oleh RSA Security Laboratories. RC-6 termasuk dalam kategori enkripsi simetri yang menggunakan kunci yang sama dalam melakukan enkripsi maupun dekripsi. RC-6 menggunakan 4 (empat) *working registers*, dan menyertakan operasi perkalian integer sebagai operasi primitif tambahan. Operasi perkalian meningkatkan penyebaran untuk tiap putarannya sehingga meningkatkan faktor keamanan, mengurangi putaran, dan meningkatkan performa hasil. Tingkat keamanan pada algoritma ini terletak pada kekuatan rotasi yang berdasarkan data, penggunaan eksklusif OR yang bergantian, fungsi modulo dan fungsi persamaan yang menggunakan rotasi yang tetap.

Perumusan Masalah

Permasalahan yang diidentifikasi dalam form login berbasis web adalah pengamanan *user-id* dan password yang dikirimkan dari client ke server agar tidak mudah didapatkan oleh hacker.

[1)] *Mahasiswa Magister Sistem Informasi Undip*
[2)] *Staff Pengajar Magister Sistem Informasi Undip*
[3)] *Staff Pengajar Magister Sistem Informasi Undip*

Solusi yang ditawarkan untuk mengatasi permasalahan ini adalah dengan melakukan enkripsi *user-id* dan password menggunakan algoritma simetri RC-6 sebelum dikirimkan. Enkripsi dengan menggunakan javascript dilakukan pada sisi client sedangkan pada sisi server dilakukan dekripsi dengan menggunakan php.

Dari penjelasan di atas, penelitian difokuskan untuk menjawab pertanyaan apakah algoritma enkripsi simetri RC-6 mampu untuk mengamankan proses login pada teknologi web ?

Batasan Masalah

Desain *web secure login* dilaksanakan dengan menggunakan bahasa pemrograman javascript pada sisi client dan pemrograman php pada sisi server. Enkripsi dan dekripsi *user-id* dan password dilakukan dengan menggunakan algoritma enkripsi simetri RC-6. Dalam hal ini penelitian dilakukan dengan batasan-batasan sebagai berikut :

- Penelitian difokuskan pada penggunaan algoritma enkripsi simetri RC-6 untuk mengamankan form login pada halaman web dan tidak membahas lebih jauh mengenai keamanan algoritma RC-6 maupun keamanan halaman web dengan metode yang lain.
- Penelitian dilakukan pada jaringan LAN, tidak pada jaringan internet.
- User-id dan password yang digunakan dibatasi maksimal 14 karakter.

Tujuan

Tujuan dari penelitian adalah merancang form login berbasis web yang aman dengan menggunakan algoritma enkripsi simetri RC-6. Implementasi rancangan dilakukan dengan menggunakan bahasa pemrograman javascript pada sisi client dan pemrograman php pada sisi server.

TINJAUAN PUSTAKA

Konsep Dasar Kriptografi

Kriptografi berasal dari bahasa Yunani yaitu *kryptos* yang artinya "yang tersembunyi" dan *graphein* yang artinya "tulisan", jadi kriptografi adalah seni dan ilmu untuk menjaga keamanan data. Dan ahlinya disebut sebagai *cryptographer. Cryptanalst* merupakan orang yang melakukan *cryptanalysis,* yaitu seni dan ilmu untuk membuka *ciphertext* menjadi *plaintext* tanpa melalui cara yang seharusnya. Data yang dapat dibaca disebut *plaintext* dan teknik untuk membuat data tersebut menjadi tidak dapat dibaca disebut *enkripsi.* Data hasil dari enkripsi disebut *ciphertext,* dan proses untuk mengembalikan *ciphertext* menjadi *plaintext* disebut *dekripsi.* Cabang matematika yang mencakup kriptografi dan *cryptanalysis* disebut *cryptology* dan pelakunya disebut *cryptologist.*

Sistem kriptografi atau *cryptosystem* adalah sebuah algoritma ditambah semua kemungkinan *plaintext*, *ciphertext* dan kunci. Dalam sistem ini, seperangkat parameter yang menentukan transformasi pengkodean tertentu disebut suatu set kunci. Proses enkripsi dan dekripsi diatur oleh satu atau beberapa kunci kriptografi. Secara umum, kunci-kunci yang digunakan untuk proses enkripsi dan dekripsi tidak perlu identik, tergantung pada sistem yang digunakan. Setiap algoritma kriptografi terdiri algoritma enkripsi (E) dan algoritma dekripsi (D). Dasar matematis yang mendasari proses enkripsi dan dekripsi adalah relasi antara dua himpunan yaitu himpunan yang berisi elemen *plaintext* dan himpunan yang berisi elemen *ciphertext*. Enkripsi dan deskripsi merupakan fungsi tranformasi antara dua himpunan tersebut. Secara umum dapat digambarkan secara matematis sebagai berikut:

Ek(P) = C (Proses Enkripsi)
Dk(C) = P (Proses Dekripsi)
Dk(E(P)) = P (Proses Dekripsi)

Dalam proses tersebut, *plaintext* disandikan dengan P dengan suatu kunci K lalu dihasilkan pesan C. Pada proses dekripsi, C diuraikan dengan menggunakan kunci K sehingga menghasilkan M yang sama dengan sebelumnya.

Algoritma RC-6

Algoritma RC-6 merupakan salah satu kandidat *Advanced Encryption Standard* (AES) yang diajukan oleh *RSA Laboratoriest* kepada NIST. Dirancang oleh Ronald L Rivest, M.J.B. Robshaw, R. Sidney dan Y.L. Yin, algoritma ini merupakan pengembangan dari algoritma sebelumnya yaitu RC5 dan telah memenuhi semua kriteria yang diajukan oleh NIST.

Algoritma RC-6 adalah versi yang dilengkapi dengan beberapa parameter, sehingga dituliskan sebagai RC-6-w/r/b, dimana parameter w merupakan ukuran kata dalam satuan bit, r adalah bilangan bulat bukan negatif yang menunjukan banyaknya iterasi selam proses enkripsi, dan b menunjukan ukuran kunci enkripsi dalam *byte*. Ketika algoritma ini masuk sebagai kandidat AES, maka ditetapkan nilai parameter w = 32, r = 20 dan b bervariasi antara 16, 24, dan 32 *byte*. RC-6-w/r/b memecah *block* 128 bit menjadi

4 buah *block* 32 bit, dan mengikuti enam aturan operasi dasar sebagai berikut :

- A + B Operasi penjumlahan bilangan integer.
- A – B Operasi pengurangan bilangan integer.
- A ⊕ B Operasi *exclusive*-OR (XOR)
- A x B Operasi perkalian bilangan integer.
- A <<< B A dirotasikan ke kiri sebanyak variabel kedua (B)
- A >>> B A dirotasikan ke kanani sebanyak variabel kedua (B)

Enkripsi RC-6

Karena RC-6 memecah *block* 128 bit menjadi 4 buah *block* 32 bit, maka algoritma ini bekerja dengan 4 buah register 32-bit A, B, C, D. *Byte* yang pertama dari *plaintext* atau *ciphertext* ditempatkan pada *byte* A, sedangkan *byte* yang terakhirnya ditempatkan pada *byte* D. Dalam prosesnya akan didapatkan (A, B, C, D) = (B, C, D, A) yang diartikan bahwa nilai yang terletak pada sisi kanan berasal dari register disisi kiri.

Algoritma RC-6 menggunakan 44 buah sub kunci yang dibangkitkan dari kunci dan dinamakan dengan S[0] hingga S[43]. Masing-masing sub kunci panjangnya 32 bit. Proses enkripsi pada algoritma RC-6 dimulai dan diakhiri dengan proses *whitening* yang bertujuan untuk menyamarkan iterasi yang pertama dan yang terakhir dari proses enkripsi dan dekripsi. Pada proses *whitening* awal, nilai B akan dijumlahkan dengan S[0], dan nilai D dijumlahkan dengan S[1]. Pada masing-masing iterasi pada RC-6 menggunakan 2 buah sub kunci. Sub kunci pada iterasi yang pertama menggunakan S[2] dan S[3], sedangkan iterasi-iterasi berikutnya menggunakan sub-sub kunci lanjutannya. Setelah iterasi ke-20 selesai, dilakukan proses *whitening* akhir dimana nilai A dijumlahkan dengan S[42], dan nilai C dijumlahkan dengan S[43].

Setiap iterasi pada algoritma RC-6 mengikuti aturan sebagai berikut, nilai B dimasukan ke dalam fungsi f, yang didefinisikan sebagai f(x) =x(2x+1), kemudian diputar kekiri sejauh lg-w atau 5 bit. Hasil yang didapat pada proses ini dimisalkan sebagai u. Nilai u kemudian di XOR dengan C dan hasilnya menjadi nilai C. Nilai t juga digunakan sebagai acuan bagi C untuk memutar nilainya kekiri. Begitu pula dengan nilai u, juga digunakan sebagai acuan bagi nilai A untuk melakukan proses pemutaran kekiri. Kemudian sub kunci S[2i] pada iterasi dijumlahkan dengan A, dan sub kunci S[2i+1] dijumlahkan dengan C. keempat bagian dari *block* kemudian akan dipertukarkan dengan mengikuti aturan, bahwa nilai A ditempakan pada D, nilai B ditempatkan pada A, nilai C ditempatkan pada B, dan nilai (asli) D ditempatkan pad C. demikian iterasi tersebut akan terus berlangsung hingga 20 kali.

Dekripsi RC-6

Proses dekripsi *ciphertext* pada algoritma RC-6 merupakan pembalikan dari proses enkripsi. Pada proses *whitening*, bila proses enkripsi menggunakan operasi penjumlahan, maka pada proses dekripsi menggunakan operasi pengurangan. Sub kunci yang digunakan pada proses *whitening* setelah iterasi terakhir diterapkan sebelum iterasi pertama, begitu juga sebaliknya sub kunci yang diterapkan pada proses *whitening* sebelum iterasi pertama digunakan pada *whitening* setelah iterasi terakhir. Akibatnya, untuk melakukan dekripsi, hal yang harus dilakukan semata-mata hanyalah menerapkan algoritma yang sama dengan enkripsi, dengan tiap iterasi menggunakan sub kunci yang sama dengan yang digunakan pada saat enkripsi, hanya saja urutan sub kunci yang digunakan terbalik.

Pembangkitan Kunci

Pengguna memasukkan sebuah kunci yang besarnya b *byte*, dimana $0 \le b \le 255$. *Byte* kunci ini kemudian ditempatkan dalam array c w-bit words L[0]…L[c-1]. *Byte* pertama kunci akan ditempatkan sebagai pada L[0], *byte* kedua pada L[1], dan seterusnya. (Catatan, bila b=0 maka c=1 dan L[0]=0). Masing-masing nilai kata w-bit akan dibangkitkan pada penambahan kunci *round* 2r+4 dan akan ditempatkan pada array S[0,…,2r+3]. Konstanta P32 = B7E15163 dan Q32 = 9E3779B9 (dalam satuan heksadesimal) adalah "konstanta ajaib"yang digunakan dalam penjadwalan kunci pada RC-6. Nilai P32 diperoleh dari perluasan bilangan biner e-2, dimana e adalah sebuah fungsi logaritma. Sedangkan nilai Q32 diperoleh dari perluasan bilangan biner ϕ-1, dimana ϕ dapat dikatakan sebagai "*golden ratio*" (rasio emas).

PHP Hypertext Preprocessor (PHP)

PHP merupakan script pemrograman yang dieksekusi menyatu dengan HTML dan bersifat *server side language*. Jadi eksekusi dari sebuah script PHP dilakukan pada server. Sedangkan

pengiriman data ke client hanya berupa tampilan HTML. PHP disimpan dalam file berekstensi .php, .php3 atau .phtml, itu tergantung dengan settingan PHP, tetapi secara umum ekstensi file PHP adalah .php. Kode PHP menyatu dengan tag – tag HTML dalam satu file. Beberapa kelebihan PHP dibandingkan dengan bahasa pemrograman yang lain yaitu :

- PHP mudah dibuat dan memiliki kecepatan akses tinggi
- PHP dapat berjalan pada web server yang berbeda dan dalam system operasi yang berbeda pula. PHP dapat berjalan di system oprasi UNIX, Windows, Windows NT, dan Macintosh
- PHP diterbitkan secara gratis
- PHP juga dapat berjalan pada web server Microsoft Personal Web Server, Apache, IIS, Xitami dan sebagainya.
- PHP termasuk bahasa yang bersifat embedded (bisa diletakkan atau ditempel pada tag HTML).

Javascript

Javascript adalah bahasa lintas-*platform* yang diperkenalkan pertama kali oleh Netscape. Javascript merupakan salah satu bahasa naskah (*scripting*) berorientasi objek. Inti Javascript mengandung sekumpulan objek, seperti Array, Date, dan Math, dan inti sekumpulan elemen bahasa seperti operator, struktur kendali, dan pernyataan.

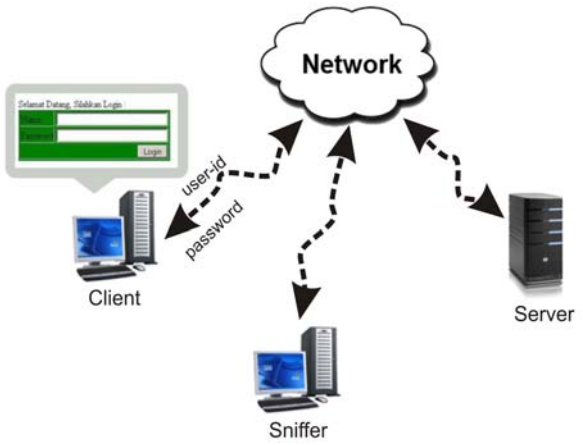
Javascript memberikan sarana untuk menjalankan aplikasi melalui Internet. Aplikasi klien berjalan di peramban seperti Netscape Navigator dan aplikasi *server* berjalan di *server* seperti Netscape Enterprise Server. Javascript dapat digunakan untuk membuat HTML dinamis yang mengolah masukan pengguna dan memelihara data menggunakan objek khusus, berkas, dan hubungan basisdata. Javascript ditemukan oleh Netscape dan digunakan pertama kali pada peramban Netscape.

METODOLOGI PENELITIAN

Kerangka Pemikiran

Deskripsi autentifikasi menggunakan form login pada teknologi web ditunjukkan pada Gambar 1.

Proses autentifikasi dimulai dengan memasukkan *user-id* dan password dalam form login yang telah ditetapkan. *User-id* dan password kemudian dikirimkan dari client ke server. Setelah menerima *user-id* dan password server melakukan autentifikasi *user-id* dan password tersebut dengan data yang tersimpan di server. Apabila *user-id* dan password valid, maka server akan memenuhi permintaan client sesuai dengan hak akses *user-id* yang dimaksud. Apabila tidak valid akan diberikan pesan ke client bahwa *user-id* dan password tidak valid.



Gambar 1. *Deskripsi autentifikasi form login dalam teknologi web*

Dalam format standar, form login akan mengirimkan *plainttext*. Misalkan *user-id* adalah "abc" dan password adalah "def", ketika menekan tombol "Login" maka data "abc" dan "def" akan dikirimkan ke server. Apabila selama proses pengiriman ini terdapat sniffer atau hacker yang mampu mendapatkan data ini, maka sniffer atau hacker tersebut bisa menggunakan data tersebut untuk melakukan login pada web yang dimaksud.

Untuk mencegah sniffer mandapatkan *plainttext* dari *user-id* dan password, dilakukan enkripsi *user-id* dan password sebelum dikirimkan ke server sehingga yang dikirimkan adalah data hasil enkripsi (*chipertext*). Server akan melakukan dekripsi *chipertext* tersebut sebelum dilakukan autentifikasi.

Alat dan Bahan Penelitian

Bahan yang dibutuhkan pada saat melakukan penelitian bersumber pada buku literatur (*text book*), paper pada jurnal terkait maupun referensi-referensi yang lain.

Perancangan dilakukan dengan menggunakan bahasa pemrograman PHP dan JAVASCRIPT. Untuk publish sistem digunakan sistem berbasis web dengan web servernya adalah APACHE.

Untuk kebutuhan perancangan sistem dan analisa hasil dibutuhkan *tools-tools* tambahan sebagai berikut :

1. *Browser* Firefox dan Internet Explorer

2. *Fiddler* v2.2.9.1
   Fiddler merupakan salah satu program pemantau sesi (*session inspector*) dengan target *browser*. Dengan perangkat lunak ini lalulintas data dari/ke *browser* dapat dipantau.
3. *Wireshark* v1.2.1
   Wireshark adalah salah satu program pengendus (*sniffer*) aliran data yang melewati jaringan. *Wireshark* ditempatkan antara server dan *browser* pada perangkat keras jaringan yang memiliki kemampuan *monitoring port*.
4. *Firebug v1.5.4*
   Firebug merupakan sebuah perangkat tambahan pada *browser mozilla* yang digunakan untuk mengukur waktu yang dibutuhkan dalam mengakes sebuah halaman web.

Desain

Perancangan dilakukan pada sisi *client* dan sisi server. Gambaran sistem pada sisi *client* ditunjukkan pada Gambar 2 sedangkan sisi server disajikan di Gambar 3.

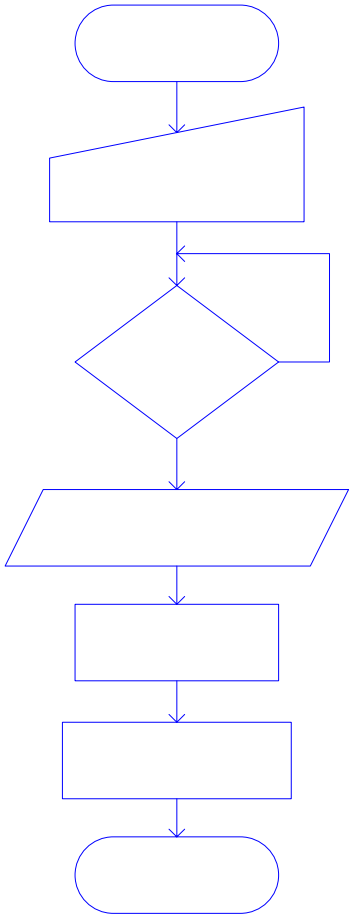
Halaman utama website menampilkan form login untuk memasukkan *user-id* dan password. Pada saat tombol login ditekan, masukan *user-id* dan password dienkripsi dengan menggunakan RC-6 sebelum dikirimkan ke server. Pada sisi server, langkah awal adalah membangun session. Selanjutnya data yang dikirimkan dari *client* akan didekripsi dengan algoritma yang sama. Hasilnya dibandingkan dengan data yang di simpan pada server. Apabila kedua data tersebut sesuai maka ditampilkan halaman web yang diminta. Pesan *error* ditampilkan apabila kedua data yang dibandingkan tidak sesuai.

HASIL PENELITIAN DAN PEMBAHASAN

Jalannya penelitian

Penelitian perancangan web secure login dengan menggunakan algoritma enkripsi simetri RC-6 ini dilakukan dengan beberapa langkah sebagai berikut :

1. Algoritma pembangkitan kunci.
2. Algoritma enkripsi simetri RC-6.
3. Algoritma dekripsi simetri RC-6.
4. Perancangan halaman login berbasis web menggunakan enkripsi RC-6.
5. Perancangan proses dekripsi user-id dan password serta autentifikasi dengan data yang disimpan pada server.



Gambar 2. *Gambaran sistem sisi client*

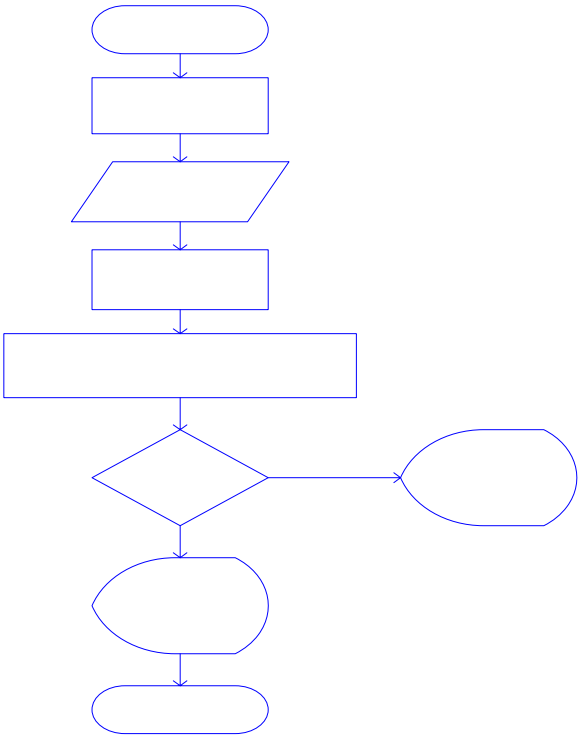
Algoritma Pembangkitan Kunci

Kunci digunakan untuk proses enkripsi dan dekripsi data pada kriptography. Dalam algoritma kriptography simetri, kunci yang digunakan dalam proses enkripsi dan dekripsi adalah sama. Dengan demikian pembangkitan kunci dalam penelitian ini dilakukan sekali untuk setiap proses enkripsi.

Pembangkitan kunci dilakukan dengan menciptakan string acak pada setiap user yang akan login. Melalui proses ini setiap user-id dan password yang dienkripsi menggunakan kunci yang berbeda. String acak ini akan ditempatkan pada array L yang merupakan barisan byte data kunci yang dibangkitkan. Setiap karakter, akan dikodekan menjadi 8 bit biner atau 1 byte. Dengan demikian jumlah data pada L sama dengan jumlah karakter pada kunci. Langkah berikutnya adalah inisialisasi kunci. Inisialisasi dilakukan sehingga tabel kunci S berisi pola bit *pseudo-random* yang tetap. Pada tahap ini, pola yang dihasilkan tidak bergantung pada kunci, tetapi bergantung pada 2 konstanta ajaib P yang memiliki nilai $b7e15163_H$ dan Q yang bernilai $9e3779b9_H$. Selanjutnya dilakukan pencampuran data S dan L untuk

mendapatkan kunci yang akan digunakan untuk enkripsi data.

Keseluruhan proses pembangkitan kunci untuk enkripsi dengan menggunakan algoritma RC-6 dapat digambarkan sesuai dengan flowchart pada Gambar 4.

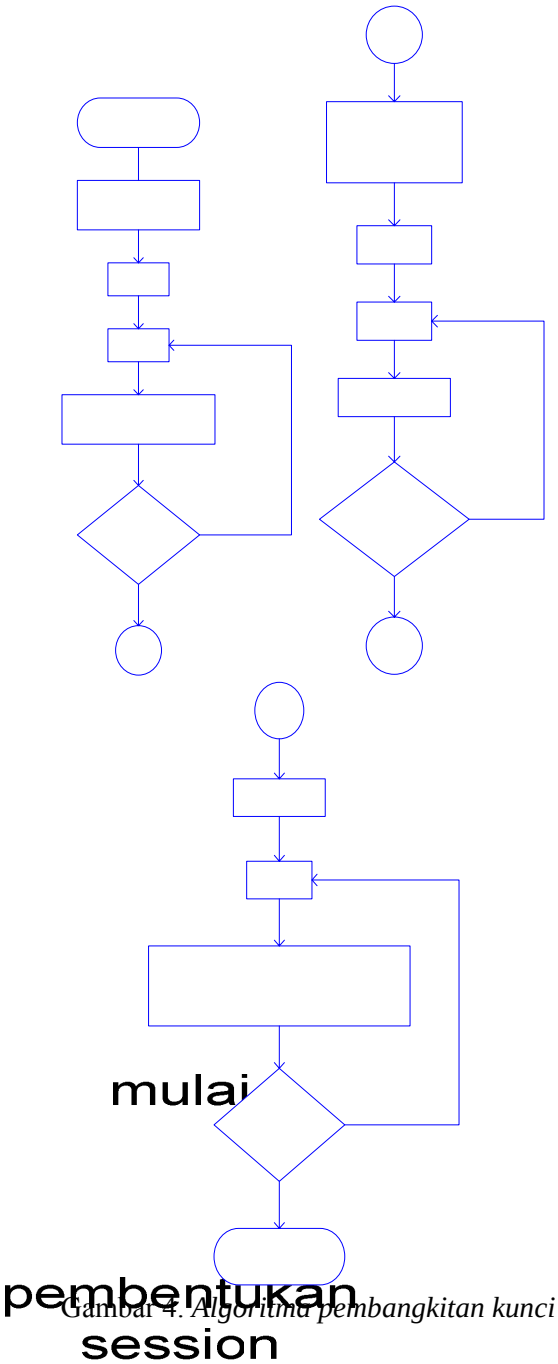


Gambar 3. *Gambaran sistem sisi server*

Algoritma Enkripsi Simetri RC-6

Fungsi enkripsi dibuat dengan menggunakan bahasa pemrograman javascript. Penggunaan bahasa pemrograman javascript dalam proses enkripsi terkait dengan proses enkripsi yang dilakukan di sisi client. Sehingga digunakan bahasa pemrograman yang dieksekusi di sisi client.

User-id dan password yang akan dienkripsi dibagi menjadi 4 blok yang nantinya akan ditempatkan pada 4 buah register A,B,C dan D. Selanjutnya empat register ini dienkripsi dengan menggunakan kunci yang telah dibangkitkan pada proses pembangkitan kunci. Pada pembangkitan kunci, masing-masing data kunci digabung menjadi satu sehingga sebelum proses enkripsi dilakukan pemisahan array kunci tersebut. Keseluruhan proses enkripsi user-id dan password menggunakan algoritma enkripsi simetri RC-6 ditunjukkan seperti pada Gambar 5.

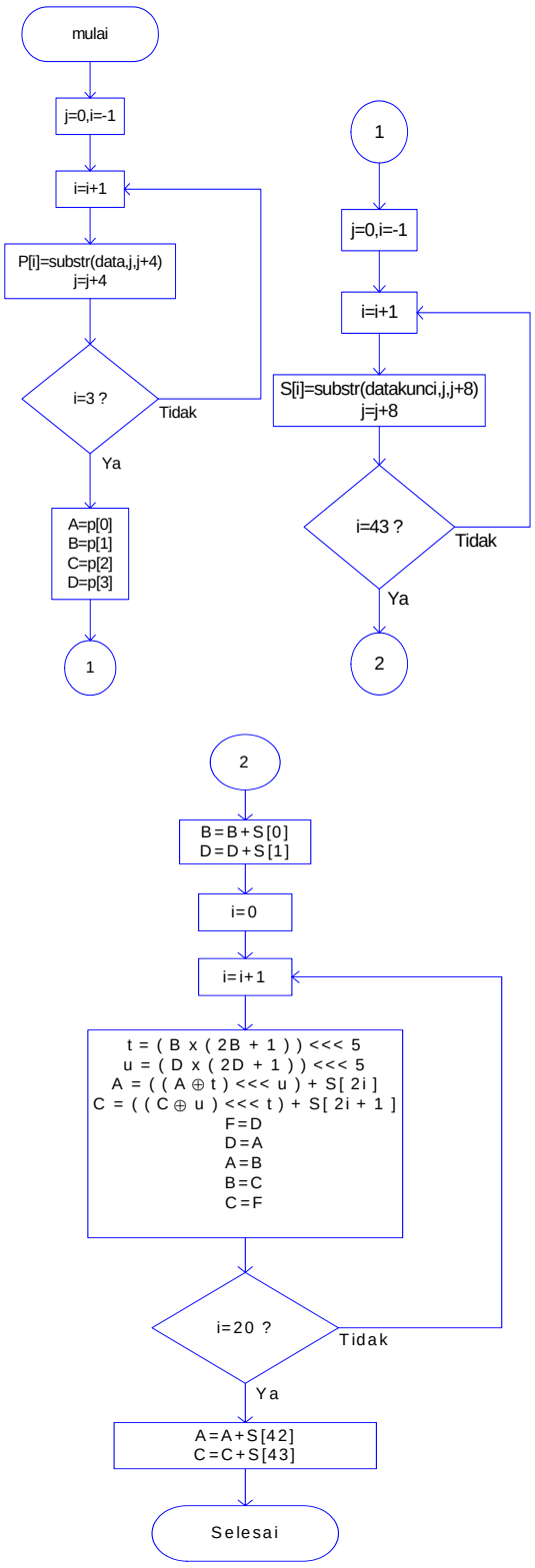


Gambar 4. *Algoritma pembangkitan kunci*

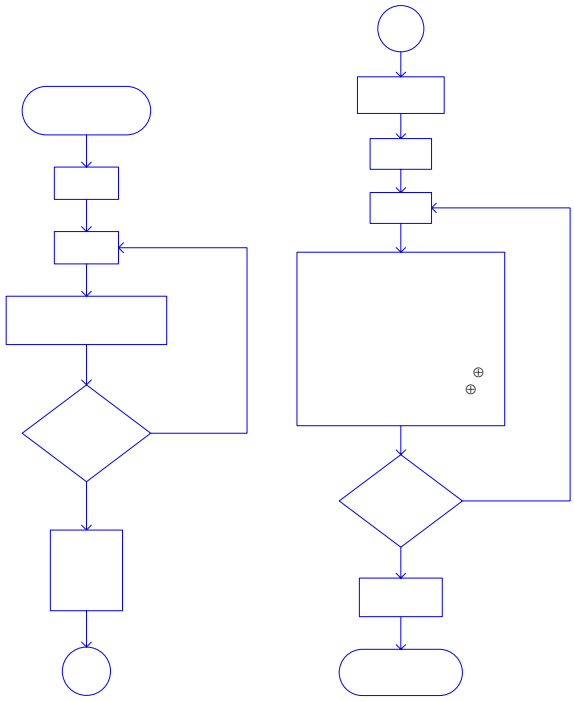
Algoritma Dekripsi RC-6

Proses dekripsi diperlukan untuk mendapatkan kembali user-id dan password yang telah dienkripsi. Proses ini dilakukan pada server sehingga bahasa yang digunakan adalah bahasa yang mendukung server side programming. Dalam hal ini digunakan bahasa pemrograman PHP. Proses dekripsi menggunakan kunci yang sama dengan kunci yang digunakan pada proses enkripsi. Pengiriman data user-id dan password dari client ke server dilakukan dengan menggunakan sebuah variabel. Oleh karenanya diperlukan proses pemisahan masing-masing register untuk proses dekripsi. Setelah diperoleh data register A,B,C dan D, dengan menggunakan

kunci yang sama dilakukan proses dekripsi data. Proses dekripsi ini dapat digambarkan menggunakan flowchart seperti pada Gambar 6.



Gambar 5. *Algoritma enkripsi simetri RC-6*



Gambar 6. *Algoritma dekripsi simetri RC-6*

Hasil Penelitian

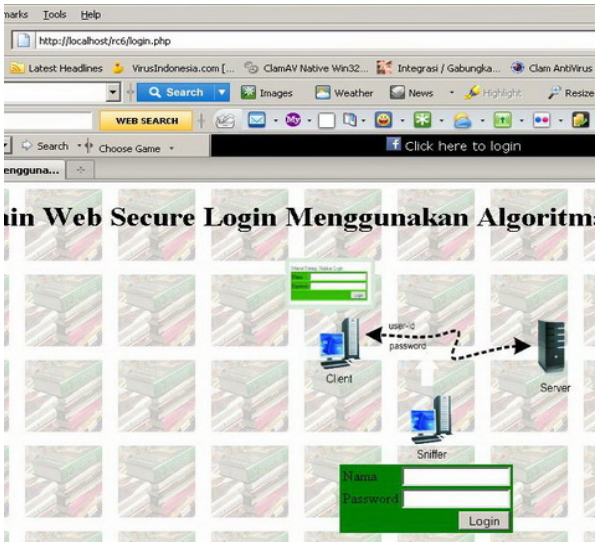
Dengan menggunakan *browser* atau program peramban, desain program ditampilkan. Dalam hal ini digunakan dua buah *browser* yaitu mozila firefox dan internet explorer. Tampilan halaman utama *website* dalam penelitian ditunjukkan pada Gambar 7.

Dengan menggunakan program *fiddler* dapat diketahui data yang ditransfer tersebut. Hal ini dilakukan dengan menjalankan program *fiddler* bersamaan dengan pengaksesan halaman *website*. Pada saat melakukan *login* pada form standar, di dapatkan data pada *fiddler* adalah nm=arkhan&y=taruna seperti pada Gambar 8. Data ini menunjukkan bahwa *form login* standar melakukan transfer data tanpa melakukan tindakan pengamanan apapun. Dengan demikian mudah diketahui bahwa *user-id* yang digunakan untuk *login* adalah arkhan dengan password taruna.

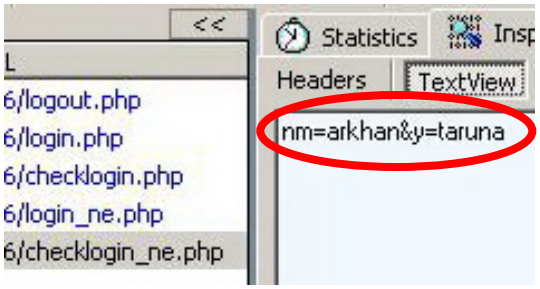
Data berbeda didapatkan pada saat *login* dengan *form* yang aman. Pada kondisi ini data yang didapatkan adalah

x=24cb4d28c6e60a306900c3500b1b7f48ad363a4
84f50f750f16bb07093866c6835a12768d7bbe470
79d69d901bf15988be0c14886026d19002418ab0a
45c46a8467701a8e891beb08aac77d02cc733c8cee
1eec870fcabd0131764f0b53220e8574cdbe8f9679
8f09b8252103d9d0e08dfb7c90881d2861023ed3f
30c607fb286822b6280a3d7330ac582c504e72e84
8f08da34892a8605034c31970d6ddd56878f89068

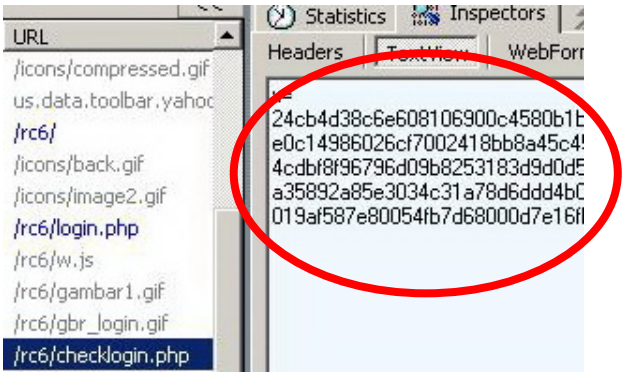
1b134d70bd2e06905f48c288&nm=00d5a3fbf000
54fb7d58000d4d232a01324e6b9e6&y=00d5a3fbf
00054fb7d58000d4d361901395b78916 seperti yang ditunjukkan pada Gambar 9. Hal ini menunjukan bahwa *user-id* dan password yang digunakan telah terenkripsi sehingga tidak diketahui data *user-id* dan password yang sebenarnya.



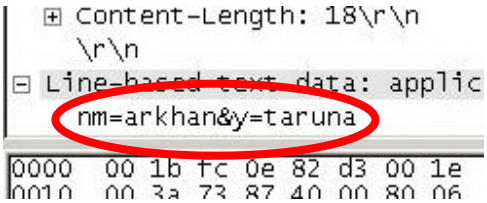
Gambar 7. *Tampilan utama website dengan form login ditambahkan dengan enkripsi*



Gambar 8. *Tampilan fiddler pada form login standar*



Gambar 9. *Tampilan fiddler pada form yang aman*

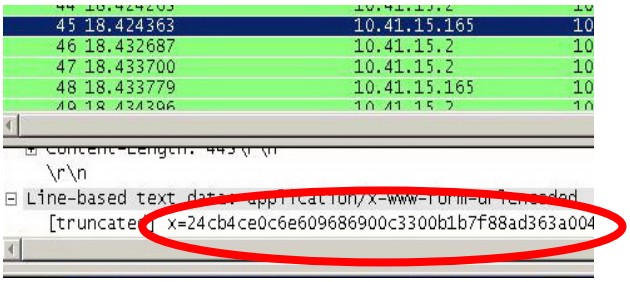


Gambar 10. *Tampilan wireshark pada form login standar*

Pengujian Menggunakan Program Sniffer *Wireshark* v1.2.1

Gambar 10 menampilkan *tampilan wireshark* saat digunakan untuk menangkap data pada *form login* standar.

Dari Gambar 10 ditunjukkan pada bagian **Line-based text data** bahwa data yang ditangkap adalah nm=arkhan&y=taruna. Data tersebut terdiri atas dua parameter nm dengan nilai arkhan dan y dengan nilai taruna yang dipisahkan dengan tanda & atau *ampersand*. Nilai arkhan dan taruna adalah *user-id* dan password yang diketikkan pada *form login* pada halaman web. Hasil berbeda diperoleh pada saat penangkapan data pada *form login* yang aman. Gambar 11 menunjukkan tampilan wireshark pada saat menangkap data yang dikirimkan dari *form login* yang aman.



Gambar 11. *Tampilan wireshark di form login yang aman*

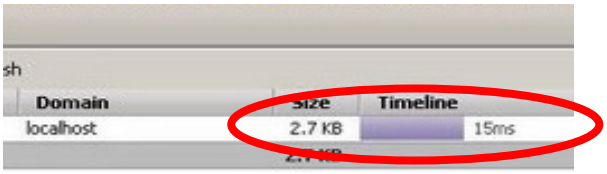
Dari tampilan tersebut terlihat bahwa data yang ditangkap oleh *wireshark* bukanlah data *user-id* dan password yang dimasukkan pada *form login* tetapi data yang udah dienkripsi.

Pengujian Kinerja *Website*

Kinerja *website* diukur dengan kecepatan akses halaman web yang dimaksud. Semakin singkat waktu yang dibutuhkan untuk mengakses sebuah halaman web semakin bagus kinerjanya.

Dalam penelitian ini pengukuran kinerja tidak dihitung dari kecepatan akses halaman *website* tetapi mengukur perubahan kecepatan akses halaman web akibat ditambahkannya proses enkripsi pada proses *login*.

Dengan menggunakan *add-on firefox firebug* waktu akses halaman web dapat diketahui. Gambar 12 merupakan hasil tampilan *firebug*.



Gambar 12 *Tampilan firebug*

Table 1. menunjukkan waktu yang dibutuhkan untuk mengakses halaman web dengan variasi *user-id* dan password pada *form login* standar dan *form* yang aman. Penambahan proses enkripsi dan dekripsi pada *form login* yang aman tidak berdampak pada bertambahnya waktu akses *website*. Pertambahan waktu ini wajar terjadi disebabkan bertambahnya jumlah file yang dipanggil dalam proses login. Dalam proses login standar, jumlah file yang dipanggil adalah 2 sedangkan pada proses login yang aman memanggil 5 buah file. Tambahan 3 file diperlukan untuk pembangkitan kunci, enkripsi dan dekripsi. Penambahan waktu rata-rata dalam percobaan ini adalah 64,67 ms (0,06467 sekon(detik)). Secara kinerja penambahan waktu akses tersebut dapat dikatakan mengurangi kinerja web namun secara visual pertambahan waktu tersebut tidak banyak berpengaruh. Penambahan waktu akses ini diimbangi dengan terpeliharanya keamanan data yang ditransmisikan melalui jaringan internet.

PENUTUP

Kesimpulan

Kesimpulan yang dapat diambil dari penelitian mengenai perancangan *form login* yang aman dengan menggunakan enkripsi simetri RC-6 adalah :

- Dengan menggunakan program pemantau sesi *fiddler* dan *sniffer wireshark* ditunjukkan bahwa halaman *form login* yang tidak menggunakan fasilitas enkripsi, *form* mengirimkan *user-id* dan password yang dimasukkan dalam bentuk *plainttext*.
- Dengan program yang sama terlihat bahwa *form login* yang menggunakan fasilitas enkripsi, *form* mengirimkan *user-id* dan password yang telah dienkripsi (*chipertext*).
- Untuk menambahkan keamanan, setiap sesi *user* dibangkitkan kunci yang digunakan untuk enkripsi dan dekripsi yang berbeda. Dengan demikian setiap *user* akan mendapatkan kunci yang berbeda setiap user tersebut login.
- Penggunaan *firebug* menunjukkan bahwa penambahan proses enkripsi pada sisi client dan dekripsi pada sisi server menambah waktu akses *website* rata-rata 64,67 ms (0,06467 sekon(detik)).. Secara visual penambahan waktu akses tidak berpengaruh dalam proses penampilan website.

Saran

Dari penelitian telah dilakukan beberapa hal yang perlu diperhatikan dalam pengembangan selanjutnya adalah :

- Penggunaan beberapa algoritma yang digabungkan dalam enkripsi *user-id* dan password agar lebih aman.
- Pengamanan *form login* adalah salah satu pengaman halaman web. Dimungkinkan untuk penggunaan metode yang lain dalam pengamanan halaman web.

Tabel 1. *Waktu akses yang dibutuhkan pada form standar dan form aman*

No	*User-id*	Password	Waktu Akses		Beda Waktu
			Standar	Aman	
1	arkhan	taruna	73 ms	141 ms	68 ms
2	arkhans	tarunaboyolali	74 ms	141 ms	67 ms
3	arkhansubari	taruna	79 ms	140 ms	61 ms
4	agus	tembalang	74 ms	145 ms	71 ms
5	herlambang	234#@!	79 ms	141 ms	62 ms
6	suryo	123suryo#@!	79 ms	141 ms	62 ms
7	andi@yahoo.com	hebat	78 ms	140 ms	62 ms
8	sempurna	solusi	73 ms	140 ms	67 ms
9	super	duper	79 ms	141 ms	62 ms

DAFTAR PUSTAKA

Anupam, V.; Mayer, A., 1998, *Secure Web Scripting,* IEEE Internet Computing, pp. 46-55.

Bellare, M.; Pointcheval, D.; and Rogaway, P., 2000, *Authenticated Key Exchange Secure Against Dictionary Attacks*, Advances in Cryptology - EUROCRYPT 2000, Proc. Int'l Conf. Theory and Application Cryptographic Techniques, LNCS 1807, Springer, pp. 139-155.

Bellovin, S.; Merritt, M., 1992, *Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks*. Proc. IEEE Symposium on Research in Security and Privacy, pp. 72-84.

Boyarsky, M., K., 1999, *Public-key Cryptography and Password Protocols: The Multi-User Case*. Proc. ACM. Computer and Communication Security, pp. 63-72.

Chakrabarti, S.; Singhal, M., 2007, *Password-Based Authentication: Preventing Dictionary Attacks*, IEEE Computer Society, pp. 68 – 74.

Halevi, S.; Krawczyk, H., 1998, *Public-key Cryptography and Password Protocols*. Proc. ACM. Computer and Communication Security, pp. 122-131.

Itani, M.; Diab, H., 2004, *Reconfigurable Computing for RC6 Cryptography*, 2004 IEEE/ACS International Conference on Pervasive Services (ICPS'04), pp. 121-127.

Prayudi, Y.; Halik, I., 2005, *Studi Dan Analisis Algoritma Rivest Code 6 (Rc6) Dalam Enkripsi/Dekripsi Data*, Seminar Nasional Teknologi Informasi 2005 (SNATI 2005), pp. 149 – 158.

Pressman, R.S., 2001, *Sofware Engineering : A Practitioner's Aproach*, McGraw-Hill.

Rivest, R., L.; Robshaw, M.J.B; Sidney, R.; Yin, Y.L., 1998, *The RC6 Block Cipher*, RCA Laboratories.

Rudianto, *Analisis Keamanan Algoritma Kriptografi RC6*, Jurusan Teknik Informatika ITB, Bandung

Stark, E.; Hamburg, M.; Boneh, D., 2009, *Symmetric Cryptography in Javascript,* 2009 Annual Computer Security Applications Conference, pp. 373-381,

Veglis, A., 2005, *PHP and SQL Made Simple,* IEEE Distributed Systems Online, pp. 4,

Yang, Y.; Zhou, J.; Weng, J.; Bao, F., 2009, *A New Approach for Anonymous Password Authentication*, 2009 Annual Computer Security Applications Conference, IEEE Computer Society, pp. 199 – 208.

Zhongying; Jiancheng, Q., 2009, *Webpage Encryption Based on Polymorphic Javascript Algorithm,* 2009 Fifth International Conference on Information Assurance and Security, pp. 327-330.