Versi Online tersedia di : https://jurnal.buddhidharma.ac.id/index.php/algor/index

JURNAL ALGOR

|2715-0577 (Online)| 2715-0569 (Print)

Artikel

Penerapan Algoritma MD5 untuk Menjaga Keamanan Terhadap *File* yang Di-*download*

Ellysha Dwiyanthi Kusuma[1]

[1] *Universitas Buddhi Dharma, Teknik Informatika, Banten, Indonesia*

SUBMISSION TRACK

Recieved: 28 Agustus 2019
Final Revision: 25 September 2019
Available Online: 30 September 2019

KEYWORD

Kriptografi, MD5, File, Download

KORESPONDENSI

E-mail: ellyshad_k@yahoo.com

A B S T R A K

Kriptografi merupakan salah satu cara yang dapat digunakan untuk menjaga keamanan data atau informasi yang terdapat dalam suatu *file* yang akan di-*download* dari pihak yang tidak bertanggung jawab. Salah satu teknik dalam kriptografi yang digunakan dalam penelitian ini yaitu metode *hash*. Metode *hash* memiliki sifat satu arah sehingga pihak yang tidak bertanggung jawab tidak akan dapat mengembalikan nilai *hash* tersebut. Penelitian ini menjaga keamanan file yang akan di-*download* oleh pengguna dengan cara memasukkan *key* yang hanya dimiliki oleh pemilik *file*. Key tersebut akan di-*generate* dengan menggunakan metode MD5. Dengan adanya *key* dalam men-*download file* dapat meningkatkan keamanan karena hanya pihak yang memiliki kewenangan yang dapat men-*download file*. Hasil pengujian *processing time* untuk men-*generate key* membutuhkan waktu tergantung dari seberapa besar *file* yang akan di-generate *key*-nya. Untuk *file* sebesar 1.000 KB membutuhkan rata-rata waktu 0,009935 detik. Sedangkan untuk *file* sebesar 10.000 KB membutuhkan rata-rata waktu 0,04836 detik.

PENDAHULUAN

Dewasa ini, perkembangan dalam bidang teknologi informasi sangat membantu dalam kehidupan sehari-hari. Hal ini terlihat dari adanya kemudahan untuk memperoleh informasi dalam bentuk *file* dengan men-*download*-nya melalui internet. Selain kemudahan dalam memperoleh informasi, terdapat permasalahan seperti berkurangnya keamanan dari pihak-pihak yang tidak bertanggung jawab untuk informasi yang bersifat rahasia maupun untuk informasi yang bersifat komersial. Kurangnya keamanan dalam hal ini dapat diatasi dengan salah satu cara yaitu teknik kriptografi. Kriptografi merupakan ilmu transmisi informasi sekaligus melindungi isi dari informasi tersebut agar tidak dimengerti oleh penerima yang tidak diinginkan sehingga keamanan informasi dapat terjamin [1]. Penelitian ini akan memanfaatkan salah satu teknik yang ada dalam kriptografi yaitu metode *hash*. Penggunaan metode *hash* dipilih karena metode ini bersifat satu arah sehingga nilai *hash* yang keluar tidak dapat dikembalikan lagi [2]. Rumusan masalah dalam penelitian ini yaitu "Bagaimana menerapkan metode *hash* MD5 untuk menjaga keamanan *file* agar hanya dapat digunakan oleh pihak yang memiliki kewenangan?".

I. METODE PENELITIAN

Menurut [3], MD5 merupakan fungsi *hash* satu arah yang dirancang sebagai pengembangan dari MD4 oleh Ron Rivest. MD5 memproses masukan ke dalam blok-blok bit sebanyak 512 bit yang kemudian dibagi ke dalam 32 bit sub blok sebanyak 16 buah. Setelah diproses maka terbentuk keluaran berupa 4 buah blok yang masing-masing berjumlah 32 bit dimana akan menjadi 128 bit yang biasa disebut nilai hash.

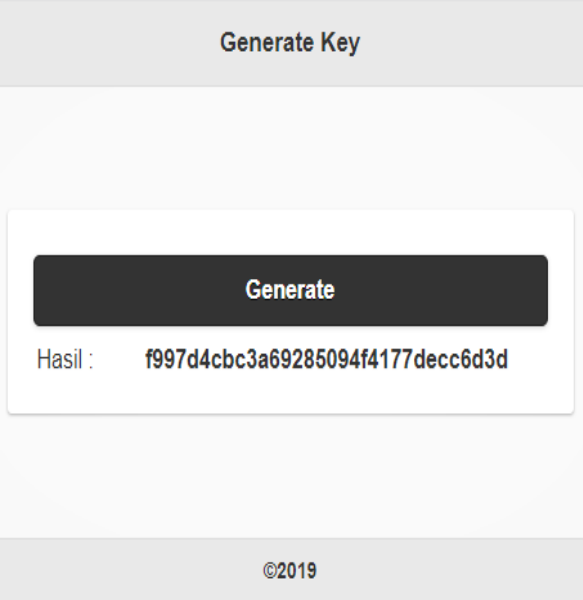
Tabel 1: Perbandingan penelitian serupa

No	Penelitian	Algoritma	Hasil
1.	Implementasi Algoritma RC4A dan MD5 untuk Menjamin *Confidentiality* dan *Integrity* pada *File* Teks, 2017 [4]	RC4A dan MD5	Keamanan data dapat bertambah karena sebelum melakukan dekripsi pesan, terlebih dahulu dilakukan verifikasi pesan.
2.	Perancangan Aplikasi Enkripsi dan Dekripsi Teks Menggunakan Algoritma *Hash* MD5 dan *Triple* DES, 2016 [5]	MD5	Kerahasiaan dari pesan teks dapat terjaga dengan adanya penerapan dari kedua algoritma tersebut.
3.	Implementasi Kriptografi Pengamanan Data pada Pesan Teks, Isi *File* Dokumen, dan *File* Dokumen Menggunakan Algoritma *Advanced Encryption Standard*, 2016 [6]	AES	Data dan informasi dapat terjaga keamanannya karena telah dilakukan pengamanan dan penyandian yang berlapis-lapis.

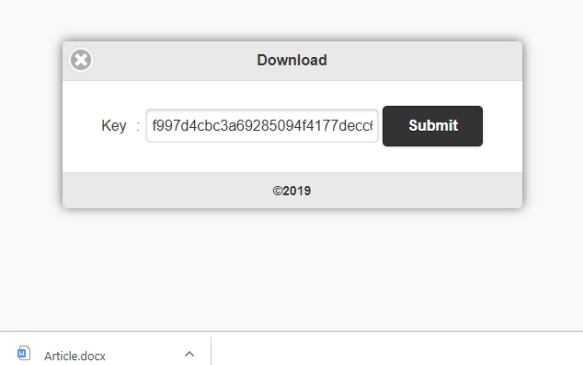
Pada penelitian ini akan dibuat dua aplikasi yaitu aplikasi untuk men-*generate key* dan aplikasi untuk men-*download file*. Aplikasi untuk men-*generate key* hanya digunakan oleh pemilik *file*. Sedangkan aplikasi untuk men-*download file* dapat digunakan oleh pihak yang membutuhkan *file*. Prosesnya dimulai dari pengguna yang mengakses aplikasi untuk men-*download file*. Kemudian pengguna akan memilih *file* mana yang dibutuhkan. Ketika akan men-*download file*, pengguna diminta untuk memasukkan *key*. Pengguna akan meminta *key* dari si pemilik *file* agar dapat men-*download file* tersebut. *Key* akan di-*generate* oleh pemilik *file* dengan aplikasi yang ada dan akan diberikan kepada pengguna. *Key* yang telah di-*generate* hanya berlaku untuk 30 menit. Apabila *key* yang dimasukkan oleh pengguna sesuai, maka *file* yang dibutuhkan dapat di-*download*. *Key* dalam penelitian ini akan dibuat menggunakan algoritma MD5. Hasil penelitian ini adalah Penerapan Algoritma MD5 untuk menjaga keamanan terhadap *file* yang di-*download*.

II. HASIL PEMBAHASAN

Gambar 1: Tampilan *List File* yang Dapat Di-*download*



Gambar 2: Tampilan *Key* yang Di-*generate* oleh Pemilik *File*



Gambar 3: Tampilan *File* yang Berhasil Di-*Download* Setelah Memasukkan *Key*

III. PENGUJIAN DENGAN SKENARIO *PROCESSING TIME*

Tabel 2: Hasil Pengujian dengan Skenario *Processing Time* (Detik) Terhadap *File* yang Berukuran 1.000 KB

Pengujian ke	Hasil
1	0,01220
2	0,00691
3	0,01097
4	0,00537
5	0,03601
6	0,00498
7	0,00633
8	0,00558
9	0,00547
10	0,00553
Rata-rata	0,009935

Tabel 3: Hasil Pengujian dengan Skenario *Processing Time* (Detik) Terhadap *File* yang Berukuran 10.000 KB

Pengujian ke	Hasil
1	0,15560
2	0,03771
3	0,03331
4	0,03453
5	0,03534
6	0,03488
7	0,03638
8	0,03418
9	0,04418
10	0,03754
Rata-rata	0,04836

IV. KESIMPULAN

Kesimpulan yang dapat diambil dari penelitian ini adalah:

1. Dengan adanya *key* yang di-*generate* dapat meningkatkan keamanan *file* yang di-*download* karena hanya pihak yang memiliki kewenangan dari pemilik *file* yang dapat men-*download file*.
2. Pada pengujian dengan perbandingan *processing time* saat men-*generate key* menghasilkan bahwa *processing time* terhadap *file* yang berukuran 1.000 KB memiliki perbedaan dengan *processing*

time terhadap *file* yang berukuran 10.000 KB. Rata-rata *processing time* yang dibutuhkan terhadap *file* yang berukuran 1.000 KB yaitu 0,009935 detik dan terhadap *file* yang berukuran 10.000 KB yaitu 0,04836 detik. Dengan demikian *processing time* yang dibutuhkan untukmen-*generate key* tergantung dari seberapa besar *file* tersebut.

REFERENSI

[1] B. Agarwal, A. B. Amara, S. Caulfield, and J. Jo, "Cryptography," 2004.

[2] E. R. Agustina, A. Kurniati, L. S. Negara, P. Minggu, and J. Selatan, "Pemanfaatan Kriptografi dalam Mewujudkan Keamanan Informasi pada e-Voting di Indonesia," *Semin. Nas. Inform. 2009 (semnasIF 2009)*, vol. 2009, no. semnasIF, pp. 22–28, 2009.

[3] Ronald L. Rivest, "RFC 1321 - The MD5 Message-Digest Algorithm (RFC1321)," 2016. [Online]. Available: http://www.faqs.org/rfcs/rfc1321.html. [Accessed: 06-Apr-2016].

[4] N. Hayati, "Implementasi Algoritma RC4A dan MD5 untuk Menjamin Confidentiality dan Integrity pada File Teks," *J. Penelit. Tek. Inform.*, vol. 1, no. April, pp. 51–57, 2017.

[5] S. MS Sanggo, A. Renaldi, and M. Norris Simbolon, "Perancangan Aplikasi Enkripsi dan Dekripsi Teks Menggunakan Algoritma Hash MD5 dan Triple DES," pp. 1–6, 2016.

[6] F. N. Pabokory, I. F. Astuti, and A. H. Kridalaksana, "Implementasi Kriptografi Pengamanan Data Pada Pesan Teks, Isi File Dokumen, Dan File Dokumen Menggunakan Algoritma Advanced Encryption Standard," *Inform. Mulawarman J. Ilm. Ilmu Komput.*, vol. 10, no. 1, p. 20, 2016.

BIOGRAPHY

Ellysha Dwiyanthi Kusuma pada tahun 2014 lulus S1 STMIK Buddhi Program Studi Teknik Informatika dan tahun 2016 lulus S2 di Universitas Budi Luhur. Pada tahun 2016 bekerja di Universitas Buddhi Dharma sebagai dosen Program Studi Teknik Informatika.