AUTENTIKASI CITRA RGB MENGGUNAKAN KOMBINASI FUNGSI HASH MD5 DAN RSA

Lekso Budi Handoko*, Chaerul Umam dan Christy Atika Sari
Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
Jl. Imam Bonjol 207, Kode Pos 50131, Semarang, Jawa Tengah
*Email: handoko@dsn.dinus.ac.id

Abstrak

Proses autentikasi pada citra sangatlah penting dalam proses berbagi gambar agar tidak terjadi pemalsuan gambar saat dikirim melalui internet. Proses autentikasi dapat dilakukan dengan memberi identitas unik kepada tiap citra. MD5 merupakan salah satu algoritma yang sering digunakan dalam proses pembentukan identitas unik yang biasa disebut message digest. Untuk melindungi identitas unik ini agar tidak dimanipulasi oleh pihak lain, penulis menggabungkannya dengan RSA untuk memberikan keamanan tambahan. RSA merupakan salah satu public crypto system yang kuat oleh karena sulitnya memfaktorkan bilangan yang besar. Pada penelitian ini, pengujian gabungan metode yang diusulkan dilakukan dengan membandingan hasil message digest tiap citra sebelum dan sesudah mengalami proses filtering dan noising serta menggunakan perhitungan SSIM untuk mengetahui tingkat kemiripan citra. Dari hasil pengujian didapatkan bahwa gabungan MD5 dan RSA memiliki sensitivitas yang tinggi terhadap perubahan kecil. Hal ini dibuktikan dengan citra yang telah mengalami manipulasi memiliki nilai SSIM 0.9678 dimana mendekati 1 yang berarti hanya mengalami sedikit perubahan. Namun hasil message digest dari citra ini memiliki perbedaan yang signifikan dibandingkan dengan message digest citra aslinya.

***Kata kunci**: Autentikasi, Citra Digital, Message Digest 5 (MD5), RSA, SSIM*

1. PENDAHULUAN

Aktivitas berbagi gambar digital melalui internet sudah menjadi kebutuhan bagi sebagian orang. Selain karena kemudahan dalam penggunaannya, biaya yang murah juga menjadi salah satu alasan digunakannya internet sebagai salah satu media dalam proses pengiriman gambar. Namun, mengirim gambar melalui internet memiliki risiko gambar tersebut dapat dimanipulasi oleh orang lain (Kusuma *et al.*, 2017). Manipulasi gambar dapat mempengaruhi informasi yang ingin disampaikan. Terkadang, penerima tidak mengetahui jika gambar tersebut telah dimanipulasi. Hal ini dapat menyebabkan terjadinya perbedaan informasi antara penerima dan pengirim. Maka dari itu perlu adanya penanda digital yang dapat menjadi identitas dari suatu *file*.

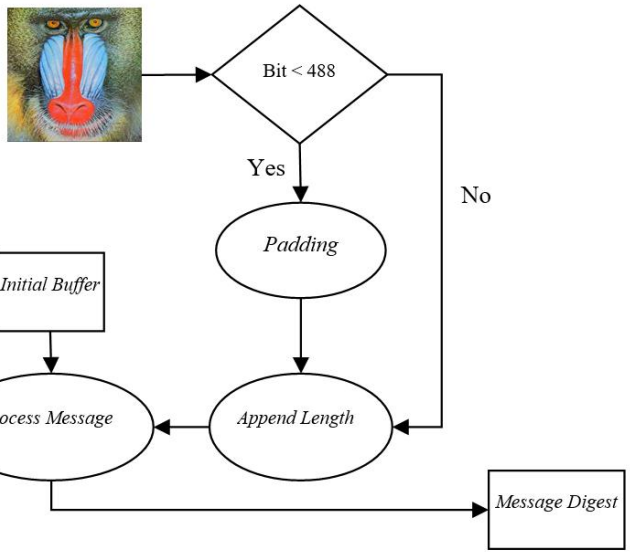
Penanda digital atau *Digital Signature* merupakan identitas unik suatu *file* yang dapat digunakan untuk proses autentikasi *file*. Proses autentikasi digunakan untuk mengetahui apakah *file* tersebut adalah *file* asli berdasarkan *digital signature*-nya. Salah satu metode yang banyak digunakan dalam proses autentikasi *file* adalah MD5 (*Message-Digest 5*). MD5 merupakan salah satu jenis *hash function* yang memiliki sensitifitas tinggi terhadap perubahan *file* orisinilnya. Bayu Seta dkk (Seta, Ridho dan Theresiawati, 2017) menggabungan MD5 dengan *Advanced Encryption Standard* (AES) dalam proses identifikasi keaslian ijazah. Pada proses nya kode ijazah akan diolah menggunakan MD5, kemudian hasilnya akan dienkripsi menggunakan AES. Pada proses autentikasi citra digital, MD5 akan mengeksekusi keseluruhan citra dan akan menghasilkan 128-bit *message digest* yang akan menjadi identitas unik dari citra tersebut (Rachmawati, Tarigan dan Ginting, 2018). Jika citra yang akan diuji keasliannya mengalami sedikit perubahan, maka *message digest* yang dihasilkan akan berbeda dengan *message digest* yang sebelumnya telah disimpan.

Menggabungkan MD5 dengan beberapa metode kriptografi lain (Damara Ardy *et al.*, 2017) dapat memberikan keamanan tambahan terhadap identitas citra yang akan ditransmisikan melalui internet. Maka dari itu, penulis mencoba untuk menggabungkan MD5 dengan RSA yang dimana RSA merupakan *public crypto system* yang kuat karena sulitnya memfaktorkan angka yang besar (Irfan, Prayudi dan Riadi, 2015). Kombinasi dua algoritma ini akan diimplementasikan pada citra berwarna yang berukuran 64 × 64 piksel untuk uji autentikasi citra tersebut.

ISBN 978-602-99334-9-9

2. *Message Digest 5* (MD5)

MD5 merupakan salah satu jenis fungsi hash searah dimana hasil nya tidak dapat dikembalikan seperti semula. MD5 akan memberikan output berupa 128-bit message digest dimana akan digunakan untuk proses autentikasi. Citra yang identik akan menghasilkan message digest yang identik pula dan seandainya citra mengalami perubahan 1 bit saja maka akan menghasilkan message digest yang berbeda (Shah, 2015). Pada prosesnya, MD5 akan melakukan beberapa langkah untuk menghasilkan 128-bit message digest yaitu:



Gambar 1. Proses pembentukan message digest pada MD5

Pada gambar 1 ditunjukan proses pembentukan *message digest* dengan menggunakan MD5. Citra akan diubah menjadi barisan biner kemudian hasil tersebut akan di-*padding* jika panjangnya kurang dari 448-bit. Lalu, hasil *padding* tadi akan di tambah 64-bit panjang pesan sehingga total bit nya adalah 512-bit. Setelah itu, proses 512-bit tadi sebanyak 16 kali dengan menggunakan *initial buffer* serta menggunakan 4 operasi putaran. Berikut adalah *initial buffer* yang umum digunakan (Rachmawati, Tarigan dan Ginting, 2018):

A = 0 1 2 3 4 5 6 7
B = 8 9 A B C D E F
C = F E D C B A 9 8
D = 7 6 5 4 3 2 1 0

Untuk melakukan operasi putarannya, MD5 akan menggunakan persamaan berikut (Seta, Ridho dan Theresiawati, 2017):

Putaran ke-1 : $F(X, Y, Z) = (X \wedge Y) \vee (\sim X \wedge Z)$
Putaran ke-2 : $G(X, Y, Z) = (X \wedge Z) \vee (Y \wedge \sim Z)$
Putaran ke-3 : $H(X, Y, Z) = X \oplus Y \oplus Z$
Putaran ke-4 : $I(X, Y, Z) = Y \oplus (X \vee \sim Z)$

Message digest yang dihasilkan akan berupa A, B, C, dan D dimana untuk membentuk *message digest* akhir hasil tersebut akan diurutkan berdasarkan abjad yaitu A hingga D.

3. RSA (Riverst-Shamir-Adleman)

RSA merupakan algoritma *public crypto system* yang dipublikasikan oleh Riverst, Shamir, dan Adleman pada tahun 1977 (Patil *et al.*, 2016). Kelebihan RSA didapat dari sulitnya melakukan faktorisasi pada bilangan kunci yang besar sehingga tahan terhadap serangan *brute-force*. Proses RSA terdiri dari pembangkitan kunci, enkripsi, dan dekripsi. Pada proses pembangkitan kunci akan menghasilkan dua buah kunci berbeda yaitu kunci publik dan kunci privat. Kunci publik digunakan dalam proses enkripsi sedangkan kunci privat digunakan dalam proses dekripsi. Proses pembangkitan kunci dapat dilihat dibawah ini:

1. Masukkan bilangan prima kedalam variabel *p* dan *q* sebagai variabel pembangkitan kunci.
2. Hitung nilai *n*, dimana *n* didapat dari $n = p \times q$.
3. Kemudian tentukan nilai $\varphi(n) = (p - 1)(q - 1)$.
4. Pilih nilai integer *e*, dimana nilai *e* merupakan bilangan yang *co-prime* dengan $\varphi(n)$.
5. Berdasarkan nilai *e*, hitung nilai *d* dengan menggunakan persamaan $(d*e) \bmod \varphi(n) = 1$.
6. Pasangan kunci publik yang digunakan dalam proses enkripsi adalah (*e*, *n*), sedangan pasangan kunci privat yang digunakan dalam proses dekripsi adalah (*d*, *n*).

Berikut adalah potongan *pseudo-code* yang digunakan dalam proses pembangkitan kunci pada RSA:

```
p= input('Masukkan nilai p = ');
q= input('Masukkan nilai q = ');
n = p*q;
teta = (p-1)*(q-1);

for zz=1:n
    e=randi([2 n]);
    if isprime(e)==1
        break;
    end
end
e= input('Masukkan nilai e = ');

val1=0;
d=0;
while (val1~=1)
d=d+1;
val1=mod(d*e,teta);
end
```

Setelah kunci dibangkitkan melalui proses diatas, maka untuk melakukan proses enkripsi digunakan persamaan berikut:

$$C = M^e \bmod n \quad (1)$$

Pada persamaan 1, *C* merupakan *ciphertext*, sedangkan *M* merupakan *message* atau pesan yang akan dienkripsi menggunakan pasangan kunci (*e*, *n*). Untuk memproduksi *C* maka *M* akan dipangkatkan dengan *e* kemudian hasilnya akan di *modulo* menggunakan *n*.

Sedangkan untuk dekripsi pada RSA digunakan persamaan dibawah ini:

$$M = C^d \bmod n \quad (2)$$

Proses dekripsi yang ditunjukan oleh persamaan 2, diketahui *ciphertext C* dan pasangan kunci privat (*d*, *n*). Untuk mendekripsikan *C* menjadi *M* maka *C* akan dieksponensialkan dengan *d* lalu kemudian di-*modulo* dengan *n.*

4. HASIL DAN PEMBAHASAN

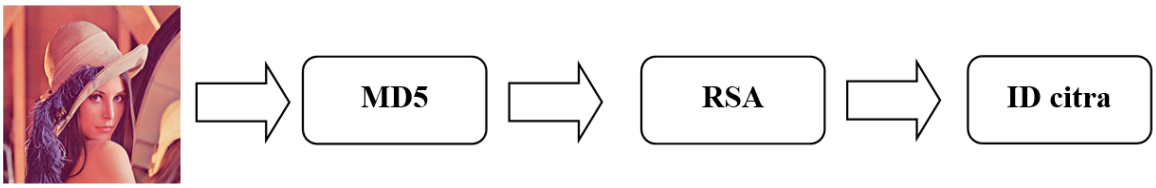
Pada penelitian ini digunakan beberapa gambar berwarna berukuran 64×64 piksel berformat .jpg sebagai data uji coba untuk mengetahui performa dari kombinasi fungsi *hash* MD5 dengan RSA.

ISBN 978-602-99334-9-9

F16 Baboon Peppers Lena

Gambar 2. Citra uji yang digunakan dalam penelitian ini

Dengan menggunakan dataset citra uji pada Gambar 2, maka langkah-langkah dalam metode yang diusulkan pada penelitian ini akan dilakukan sesuai dengan Gambar 3:



Gambar 3. Tahapan metode yang diusulkan

Pada Gambar 3 ditunjukan tahapan-tahapan yang dilakukan dalam penelitian ini. Pada tahap awal, citra akan diolah menggunakan MD5 dan akan menghasilkan 128-bit *message digest*. Kemudian, hasil tersebut akan direkonstruksi menjadi 32-bit hexadesimal dimana tiap 4-bit *message digest* akan diubah menjadi 1-bit hexadesimal. Setelah itu, 32-bit hexadesimal akan direkonstruksi ulang menjadi 16-bit angka desimal, dimana setiap 2-bit hexadesimal akan di-*convert* menjadi bilang desimal untuk kemudian dienkripsi dengan menggunakan algoritma RSA.

Untuk mengetahui kualitas *message digest* yang dijadikan sebagai *digital signature* pada citra, kami telah melakukan beberapa pengujian yaitu waktu yang diperlukan untuk menghasilkan *final message digest* tiap citra, melihat tingkat kemiripan *message digest*-nya, serta menggunakan SSIM (*Structure Similarity Index)* untuk mengetahui tingkat kemiripan citra setelah mengalami berbagai jenis manipulasi seperti *gaussian filter*. Berikut adalah persamaan yang digunakan untuk menghitung SSIM:

$$SSIM(A,B) = \frac{(2\mu_A\mu_B + c_1)(2\sigma_{AB} + c_2)}{(\mu_A^2 + \mu_B^2 + c_1)(\sigma_A^2 + \sigma_B^2 + c_2)} \quad (6)$$

Dimana :

μ_A dan μ_B = rata-rata dari citra A dan citra B

σ_{AB} = Covarian citra A terhadap citra B

σ_A^2 = varian dari citra A

σ_B^2 = varian dari citra B

$c_1 = (k_1 L)^2; c_2 = (k_2 L)^2$

L adalah *dynamic range* citra ($2^{bit} - 1$) dengan nilai *default*-nya adalah k_1= 0.01 dan k_2= 0.03. Hasil uji coba metode yang diusulkan menggunakan dataset citra pada Gambar 2 dipresentasikan pada Tabel 1 dimana parameter yang digunakan dalam proses pembangkitan kunci RSA adalah *p* = 7 dan *q* = 13. Hasil pembangkitan kunci RSA terdiri dari *e* = 13 dan *d* = 91.

Tabel 1. Pengujian autentikasi hasil MD5 & RSA dengan beberapa jenis manipulasi

Citra	MD5 & RSA (Desimal)	Jenis Manipulasi	MD5 & RSA (Desimal)	SSIM	Otentik?
F16	12 24 2 80 46 44 4 36 2 19 52 47 71 33 3 88	-	12 24 2 80 46 44 4 36 2 19 52 47 71 33 3 88	1	Ya
		Gaussian filter	1 25 42 54 89 71 12 67 39 18 84 44 9 48 45 78	0.6852	Tidak
		Salt peper 0.01	41 73 53 28 63 33 75 20 53 34 46 32 3 2 64 60	0.8260	Tidak
		Salt peper 0.05	65 31 61 29 47 18 1 34 27 26 28 82 69 50 70 16	0.4703	Tidak
Baboon	21 18 10 24 73 46 90 64 36 14 86 84 24 22 60 40	-	21 18 10 24 73 46 90 64 36 14 86 84 24 22 60 40	1	Ya
		Gaussian filter	43 8 23 57 14 84 81 65 12 51 24 52 62 71 45 82	0.7866	Tidak
		Salt peper 0.01	17 26 67 79 68 13 46 7 67 82 79 12 38 64 29 52	0.8945	Tidak
		Salt peper 0.05	0 86 71 45 33 87 71 53 54 41 43 24 66 24 78 29	0.6836	Tidak
Peppers	64 49 53 36 2 67 25 45 23 88 43 0 28 84 9 49	-	64 49 53 36 2 67 25 45 23 88 43 0 28 84 9 49	1	Ya
		Gaussian filter	14 52 88 7 33 70 33 60 9 43 80 69 41 57 31 86	0.8505	Tidak
		Salt peper 0.01	38 63 65 52 54 33 4 26 37 6 31 12 90 83 25 47	0.9678	Tidak
		Salt peper 0.05	90 85 88 71 86 35 26 34 44 32 90 36 59 73 25 13	0.8258	Tidak
Lena	43 63 21 5 27 29 64 25 64 79 53 57 54 1 62 3	-	43 63 21 5 27 29 64 25 64 79 53 57 54 1 62 3	1	Ya
		Gaussian filter	83 90 11 23 18 40 56 79 63 37 57 72 86 81 10 21	0.8794	Tidak
		Salt peper 0.01	40 4 59 79 11 84 63 54 71 57 1 51 17 56 73 20	0.9567	Tidak
		Salt peper 0.05	47 90 0 89 48 87 77 1 82 67 32 34 66 33 30 20	0.7917	Tidak

Dari Tabel 1 dapat dilihat bahwa hasil dari MD5 dan RSA berupa deret bilangan desimal yang merepresentasikan identitas unik tiap citra. Dalam pengujian ini, citra asli akan dibandingkan dengan citra asli tanpa manipulasi, kemudian citra yang sudah dimanipulasi dengan menggunakan *gaussian filter*, serta *salt and pepper noise* dengan intensitas *noise* 0.01 dan 0.05.

Dari hasil pengujian yang ditunjukan oleh Tabel 1, dapat dicermati bahwa setelah citra mengalami manipulasi oleh proses *filtering* dan *noise*, *message digest* citra tersebut juga berubah. Kasus lain yang ada citra peppers yang telah diberi *salt and pepper noise* dengan intensitas 0.01 dapat menghasilkan nilai SSIM sebesar 0.9678 dimana dengan nilai SSIM tersebut hampir mendekati 1 menandakan bahwa citra tersebut mengalami sedikit perubahan, nilai *message digest*-nya sangat berbeda dibandingkan dengan citra aslinya. Maka dapat disimpulkan bahawa gabungan antara MD5 dan RSA pada penelitian ini dapat menghasilkan *message digest* atau *digital signature* yang sensitif terhadap perubahan, dimana perubahan sekecil apapun akan mengubah seluruh *message digest* atau *digital signature* citra tersebut.

5. KESIMPULAN

Autentikasi citra digital merupakan salah satu cara dalam mengurangi terjadinya pemalsuan informasi. Menggabungan MD5 dan RSA dalam proses pembentukan identitas unik citra telah dilakukan dalam penelitian ini. Hasil uji didapatkan bahwa gabungan metode ini dapat menghasilkan metode pembentukan *message digest* yang sangat sensitif terhadap perubahan kecil. Hal ini dibuktikan pada citra yang telah mengalami manipulasi *noise* dengan nilai SSIM 0.9678 dimana nilai tersebut mendekati 1 yang berarti hanya mengalami perubahan yang sangat kecil dapat menghasilkan *message digest* yang sangat berbeda.

ISBN 978-602-99334-9-9

DAFTAR PUSTAKA

Damara Ardy, R. *et al.* (2017) "Digital image signature using triple protection cryptosystem (RSA, Vigenere, and MD5)," in *2017 International Conference on Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS)*. IEEE, hal. 87–92. doi: 10.1109/ICON-SONICS.2017.8267827.

Irfan, P., Prayudi, Y. dan Riadi, I. (2015) "Image Encryption using Combination of Chaotic System and Rivers Shamir Adleman (RSA)," *International Journal of Computer Applications*, 123(6), hal. 11–16.

Kusuma, E. J. *et al.* (2017) "An Imperceptible LSB Image Hiding on Edge Region Using DES Encryption," *International Conference on Innovative and Creative Information Technology (ICITech)*, (March), hal. 1–5. doi: 10.1109/INNOCIT.2017.8319132.

Patil, P. *et al.* (2016) "A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish," *Procedia Computer Science*. Elsevier Masson SAS, 78(December 2015), hal. 617–624. doi: 10.1016/j.procs.2016.02.108.

Rachmawati, D., Tarigan, J. T. dan Ginting, A. B. C. (2018) "A comparative study of Message Digest 5(MD5) and SHA256 algorithm," *Journal of Physics: Conference Series*, 978(1), hal. 12116. doi: 10.1088/1742-6596/978/1/012116.

Seta, H. B., Ridho, M. M. dan Theresiawati (2017) "Kombinasi Algoritma Advanced Encryption Standard (AES) Dan Hash Untuk Mengidentifikasi," *Seminar Nasional Teknologi Informasi dan Multimedia 2017*, hal. 7–12.

Shah, D. (2015) "Digital Security Using Cryptographic Message Digest Algorithm," *International Journal of Advance Research in Computer Science and Management Studies*, 3(10), hal. 2321–7782.