Mendeteksi Keaslian Data Menggunakan Kombinasi *Message Digest*-5 dengan RSA *Public Key Algorithm*

Agung Purnomo Sidik
[1]Fakultas Sains & Teknologi, Universitas Pembangunan Panca Budi
Jalan Gatot Subroto Km. 4,5 - Kota Medan - Sumatera Utara
agung@dosen.pancabudi.ac.id

Abstrak

Penelitian ini dilakukan untuk mengetahui tingkat validitas atau keaslian data. Hal ini untuk menjamin bahwa informasi pada data belum berubah dan tidak akan merugikan penerima data. Jika terjadi perubahan data, maka penerima dapat mengindentifikasinya dengan cepat, mudah, dan akurat. Metode ini mampu memberikan keamanan pada keaslian data untuk mencegah terjadinya hal buruk akibat perubahan data secara ilegal. Strategi ini menggabungkan algoritma MD5 untuk membentuk *message digest* dan algoritma RSA sebagai *public key* untuk menghasilkan sebuah *digital signature* yang unik dari data. *Digital signature* tersebut berkorelasi dengan data secara biner sehingga perubahan sebuah biner pada data akan mengubah secara total bentuk dari *digital signature*. Algoritma yang diajukan diimplementasikan dengan bahasa PHP. Hasil analisis membuktikan bahwa perubahan sekecil apa pun (bahkan hanya satu perubahan biner) pada data dapat dengan mudah dideteksi dengan hasil yang sangat akurat dan waktu proses yang cepat bahkan untuk file yang berukuran besar. Hal ini dikarenakan semakin besar ukuran file, maka rata-rata waktu proses per kilo byte akan semakin cepat. Rata-rata waktu proses untuk file berukuran 581,1MB hanya sebesar 0,00122 ms/kb. Rancangan algoritma pembangkitan *digital signature* pada penelitian ini juga terbukti sangat aman karena memiliki 2^{128} kombinasi *message digest* yang mungkin terjadi. Jika diasumsikan seorang *intruder* dapat melakukan operasi percobaan 1 triliun operasi per detik, maka dibutuhkan waktu hingga $2^{128}/10^{12}$ detik atau 10.790,283 triliun tahun lamanya untuk menemukan pasangan *message digest* yang sesuai secara *brute force*. Hal ini menjadi salah satu alasan model rancangan algoritma ini menjadi sangat aman untuk diimplementasikan.

Kata kunci: *Digital signature*, *Message digest*, MD5, RSA, Keaslian data

1. PENDAHULUAN

Jaminan keamanan untuk keaslian data yang dikirimkan sangat penting dalam komunikasi. Penerima membutuhkan jaminan bahwa data tidak mengalami perubahan saat dikirimkan ke penerima [1]. Oleh karena itu, keaslian data yang dikirim harus benar-benar mampu dijaga ketat, dan harus ada jaminan bahwa penerima menggunakan data yang benar (asli belum ada perubahan setelah dikirimkan) [2]. Salah satu alternatif untuk mengetahui keaslian data adalah dengan menerapkan teknik *digital signature* [3][4][5].

Digital signature merupakan salah satu strategi yang dapat digunakan untuk mendukung keaslian data, sehingga dimungkinkan untuk dapat memastikan bahwa data yang diperoleh adalah asli atau palsu [6]. Strategi ini dapat mencegah penerima data menggunakan data palsu hasil modifikasi [5]. Setiap data memiliki susunan biner yang unik yang dapat dipadatkan ke dalam sebuah *message digest* dengan panjang yang tetap menggunakan algoritma MD5, sehingga sedikit saja perubahan yang terjadi pada susunan biner dari data akan mengubah secara total nilai *message digest* yang dihasilkan [5][6].

Pembuatan data yang dilengkapi dengan *digital signature* bertujuan untuk memberikan perlindungan terhadap keaslian data, dimana keaslian data akan menjadi prioritas utama pada saat dikirimkan [7]. Dengan *digital signature*, ada jaminan bahwa data yang diterima belum diubah, autentik, dan masih sama seperti saat dikirimkan oleh pengirim [1][8].

DOI: 10.34128/jsi.v8i1.399 *Received: 4 April 2022*
Accepted: 27 Mei 2022

Penelitian ini bertujuan untuk merancang model algoritma untuk menghasilkan *digital signature* dari data untuk mendeteksi keaslian data secara akurat. Algoritma ini ditargetkan dapat diterapkan pada semua jenis file baik file dokumen, text, citra, audio, maupun video.

2. TINJAUAN PUSTAKA

Banyak dari penelitian terdahulu yang melakukan penelitian tentang *digital signature* hanya dengan menggunakan algoritma kunci publik tanpa algoritma *message digest* [5][6][7]. Beberapa penelitian lain juga membangkitkan *digital signature* hanya dengan algoritma *message digest* sehingga masih memungkinkan intruder untuk memodifikasi isi file tanpa ketahuan [10][12][13]. Pada penelitian ini, akan dikombinasikan algoritma *message digest* MD5 dan algoritma kunci publik RSA untuk menghasilkan algoritma yang lebih baik untuk *digital signature* yang lebih baik.

2.1. Digital Signature

Digital signature merupakan sebuah penanda unik untuk setiap data atau file yang selalu berbeda antara satu dengan lainnya selama data atau file tersebut tidak 100% sama [1]. *Digital signature* dapat dimanfaatkan untuk memeriksa keaslian data atau file dengan mengidentifikasi penanda unik tersebut [8]. Proses pemeriksaan keaslian data dapat dilakukan dengan membandingkan penanda unik awal sebelum data disimpan atau dikirim dengan penanda unik akhir saat data akan digunakan atau setelah data diterima [4]. Jika penanda awal dan akhir ini 100% sama, maka dapat dikatakan 100% data tersebut masih terjaga keasliannya dan sama sekali belum dimodifikasi atau diubah [1]. *Digital signature* sangat berbeda dengan tanda tangan elektronik. Tanda tangan elektronik hanya memberikan jaminan bahwa suatu dokumen digital terjamin keabsahannya dan telah benar ditandatangani dan diketahui oleh pihak yang membuat dokumen tersebut, serta telah tercatat secara digital dan dapat dicek keberadaannya dokumen tersebut secara digital [9]. Sehingga tanda tangan elektronik menjamin keabsahan dokumen digital, sedangkan *digital signature* menjamin keaslian dari file dengan segala bentuk format file tanpa terkecuali [9] [1] [2] [7].

2.2. *Message Digest* 5 (MD5)

Ronald L Rivest merupakan pengembang algoritma *Message Digest* 5 yang dikenal dengan algoritma MD5 pada tahun 1991 [10]. Algoritma ini merupakan pengembangan dari algoritma MD4 yang telah terbukti terdapat kelemahan fatal di dalamnya [11]. Algoritma MD5 merupakan algoritma fungsi hash yang berfungsi untuk mengekstraksi setiap file menjadi 128-bits unik sebagai intisari file yang selalu berbeda antara satu file dengan file lainnya jika file tersebut tidak 100% identik [12].

Algoritma MD5 memiliki 4 tahapan pemrosesan, yaitu: penambahan bit-bit pengganjal, penambahan nilai panjang semula, inisialisasi penyangga (buffer) MD, dan pengolahan pesan pada blok 512-bit [13].

2.3. RSA *Public Key Algorithm*

RSA merupakan singkatan dari tiga orang pengembang algoritma ini, yaitu Rivest, Shamir, dan Adleman pada tahun 1970-an [14]. Algoritma RSA yang juga dikenal dengan RSA *public key algorithm* termasuk ke dalam algoritma kunci publik yang memiliki sebuah kunci rahasia (*private key*) yang digunakan untuk proses dekripsi dan sebuah kunci publik tidak rahasia (*public key*) yang digunakan untuk proses enkripsi [15].

Proses pembangkitan kunci dengan algoritma RSA membutuhkan dua buah bilangan prima [14]. Inti kekuatan dari algoritma ini adalah sulitnya pemfaktoran sebuah bilangan bulat yang besar menjadi dua buah bilangan prima yang membuat algoritma ini tetap aman jika menggunakan dua buah bilangan prima yang sangat besar [16].

3. METODOLOGI PENELITIAN

3.1. Analisis Masalah

Pada tahap ini, analisis yang mendalam dilakukan terhadap masalah, dan dicari solusi atas permasalahan tersebut. Dalam penelitian ini, masalah yang dianalisis adalah perlunya jaminan atas keaslian data dalam proses komunikasi agar penerima data tidak menggunakan data palsu atau yang telah dimodifikasi oleh penyusup yang dapat membahayakan penerima data [1][2][3].

3.2. Pengumpulan Data

Pengumpulan data merupakan tahapan untuk mengumpulkan data yang berkaitan dengan masalah yang dianalisis dan proses pemecahannya [6]. Data yang digunakan untuk pengujian terdiri dari berbagai jenis data dengan ekstensi file yang berbeda-beda dan ukuran file yang juga berbeda-beda.

3.3. Analisis Penyelesaian Masalah

Berdasarkan masalah yang diuraikan, selanjutnya dilakukan analisis terhadap teknik pemecahan masalah yang ada sehingga masalah yang diteliti dapat ditemukan pemecahan masalahnya. Proses analisis pemecahan masalah pada penelitian ini dibagi menjadi beberapa proses, yaitu:

1. **Analisis Input:** Pada sub-tahap ini, dilakukan analisis dari input-an yang dibutuhkan untuk proses pemecahan masalah.
2. **Analisis Proses:** Pada sub-tahap ini, proses pemecahan masalah dari masalah yang diteliti dianalisis. Tools yang digunakan adalah XAMPP dan bahasa PHP yang digunakan untuk membangun aplikasi dari proses pemecahan masalah yang diajukan.
   Tahap analisis proses dibagi menjadi tiga sub-bab proses, yaitu:
   a. ***Key Generation*.** Proses ini menganalisis cara menghasilkan kunci yang digunakan untuk menghasilkan tanda tangan digital dan menguji keaslian data. Proses pembentukan atau pembangkitan algoritma kunci RSA, yaitu [14]:
      1) Pilih dua bilangan prima acak besar, p dan q [17]
      2) Hitung modulus sistem p dan q dengan rumus n = p * q [16]
      3) Cari Totient Φ (n) dari p dan q dengan rumus: Φ (n) = (p-1) (q-1) [13]
      4) Pilih bilangan e secara acak sebagai kunci enkripsi dengan syarat: [10]
         $1 < e < \Phi(n)$, gcd (e, Φ (n)) = 1
      5) Lengkapi rumus berikut untuk menentukan kunci dekripsi d [6].
         $d \equiv e^{-1} \pmod{\Phi(n)}$
         dimana rumus tersebut setara dengan:
         $e * d \equiv 1 \pmod{\Phi(n)}$, where $0 \leq d \leq n$ [15]
         Hasil pembangkitan kunci mengikuti aturan berikut
         Private key = (d, n), Sangat rahasia, dan hanya penerima pesan yang boleh mengetahuinya [11].
         Public key = (e, n), Tidak rahasia dan dapat didistribusikan secara bebas [18].
   b. **Membangkitkan *Message Digest*.** Proses ini berfungsi untuk membangkitkan *message digest* dari data dengan menggunakan algoritma MD5 sehingga dihasilkan 32 bilangan heksadesimal yang unik sebagai *message digest* dari data.
   c. **Membentuk *Digital Signature*.** Proses ini berfungsi untuk membentuk *digital signature* dari data yang unik dan mampu secara akurat mengidentifikasi keaslian data dan sangat sensitif terhadap perubahan data [1] [2]. Kepekaan yang tinggi terhadap perubahan data ini sangat diperlukan [3]. Jika penyadap hanya mengubah sedikit data atau bahkan hanya mengubah 1 bit data, maka keaslian data dapat diidentifikasi secara akurat [1][4]. Pengirim akan menghasilkan data untuk menjadi *message digest*, kemudian *message digest* dienkripsi dengan algoritma RSA menggunakan kunci privat. Penelitian ini melakukan enkripsi menggunakan kunci privat, bukan kunci publik yang umumnya digunakan untuk enkripsi pada RSA. Sehingga formula pada proses enkripsi pada penelitian ini menjadi

p-ISSN: 2460-173X
e-ISSN: 2598-5841

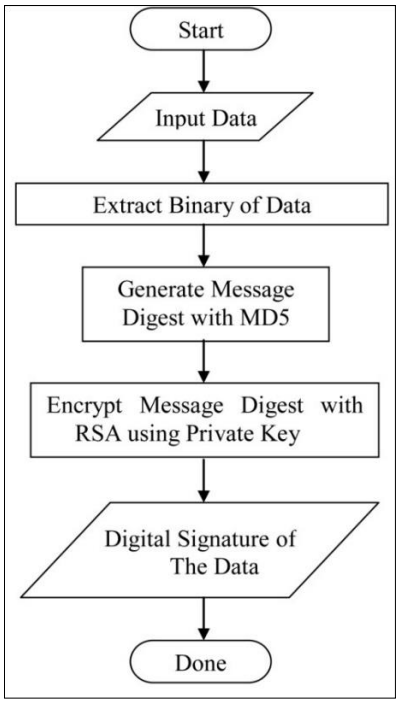
$$C_i = P_i^d \bmod n \quad (1)\ [6]$$

Hal ini bertujuan agar hanya pengirim pesan yang dapat menghasilkan *digital signature* dari data yang dikirimkan karena *private key* yang bersifat sangat rahasia. Misalkan selama proses pengiriman, penyusup memodifikasi atau membuat data baru, penyusup tidak akan menghasilkan *digital signature* yang sesuai karena penyusup tidak memiliki *private key*.

d. Menguji Keaslian Data. Proses ini menganalisis bagaimana menguji keaslian data yang diterima oleh penerima data. Penerima akan menguji keaslian data berdasarkan *digital signature* dengan mendekripsi *digital signature* menggunakan *public key*. Proses deskripsi pada penelitian ini menggunakan kunci publik, bukan kunci privat. Oleh karena itu, setiap penerima dapat memeriksa keaslian file tersebut. Formula untuk proses dekripsi dalam penelitian ini menjadi:

$$P_i = C_i^e \bmod n \quad (2)\ [6]$$

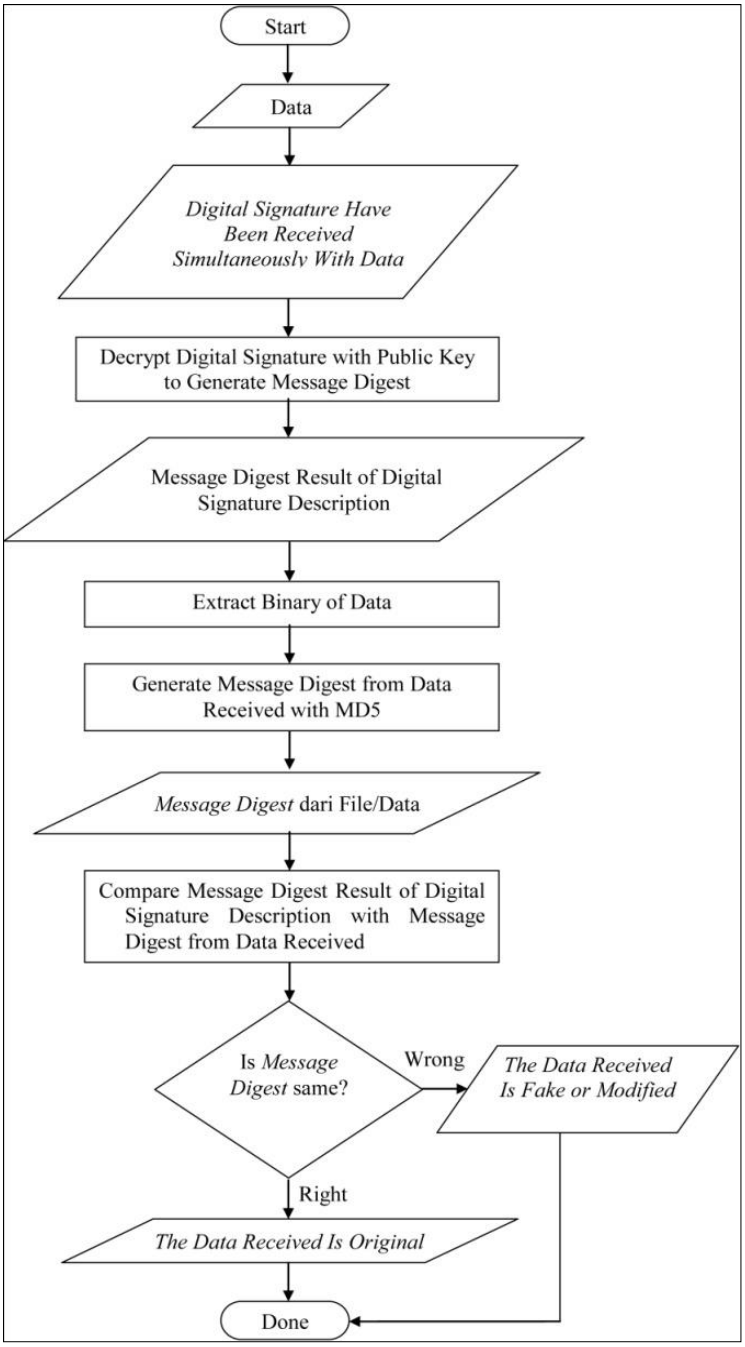
3. **Analisis Hasil:** Pada tahap ini, keluaran harus berupa pemecahan masalah untuk keaslian data. Hasilnya adalah *digital signature* yang bersifat unik.
4. **Perancangan Model *Digital Signature*.** Setelah tahapan analisis input, analisis proses, dan analisis *output*, pada tahap ini dirancang model *digital signature* baru untuk meningkatkan keamanan pada keaslian data secara lebih akurat. Oleh karena itu, pada tahap ini akan dihasilkan model *digital signature* baru yang dihasilkan dari analisis pemecahan masalah yang ada. Model menjelaskan bahwa pengirim akan menghasilkan *digital signature* dari file, sedangkan penerima akan menguji keaslian file. MD5 digunakan untuk menghasilkan intisari file atau *message digest* dari file [10][11][12]. Algoritma RSA berguna untuk mengacak *message digest* yang dibuat untuk mengamankan *message digest* agar tidak dapat diubah [6]. Proses yang dilakukan dalam model pembangkitan *digital signature* pada penelitian ini dapat dilihat pada *flowchart* berikut:



Gambar 1. *Flowchart* dari Model Pembangkitan *Digital Signature*

Proses pengujian keaslian data dengan *digital signature* pada penelitian ini dapat dilihat pada *flowchart* berikut:

p-ISSN: 2460-173X
e-ISSN: 2598-5841



Gambar 2. *Flowchart* dari Proses Pengujian Keaslian Data

3.4. Implementasi dan Pengujian Model *Digital Signature*

Dalam mengimplementasikan desain model *digital signature* yang dirancang, dibutuhkan beberapa instrumen untuk pengujian. Instrumen tersebut adalah: hardware (seperangkat komputer dengan spesifikasi RAM 8GB dan Core™ i7-8750H), aplikasi yang dibangun berdasarkan algoritma yang diusulkan, Browser, XAMPP, dan file dari beberapa ekstensi file dengan ukuran yang berbeda-beda.

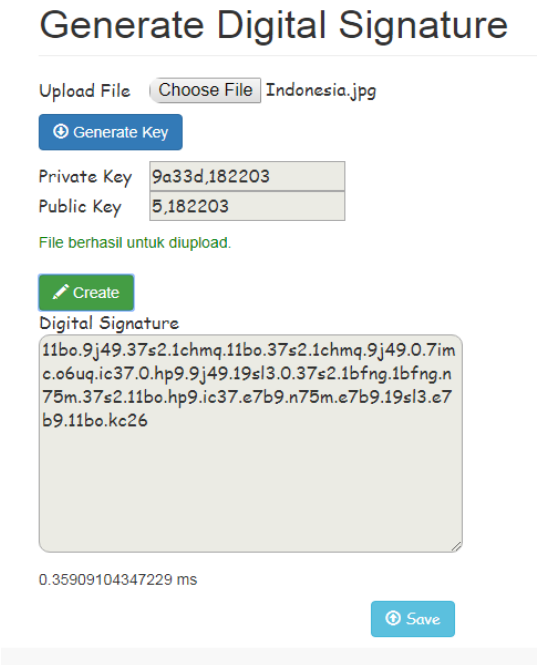
4. PEMBAHASAN

Pengujian dilakukan dengan berbagai jenis file. Pengujian pertama adalah menguji file *image* yang diberi nama Indonesia.jpg seperti yang terlihat pada gambar berikut:

p-ISSN: 2460-173X
e-ISSN: 2598-5841

Gambar 3. File Indonesia.jpg

Proses pembentukan *digital signature* dari file Indonesia.jpg dapat dilihat pada gambar berikut:



Gambar 4. Pembangkitan *Digital Signature* Menggunakan Aplikasi yang Dibangun

Hasil pembangkitan *digital signature* dari file Indonesia.jpg menggunakan aplikasi yang dibangun berdasarkan model *digital signature* yang dirancang:

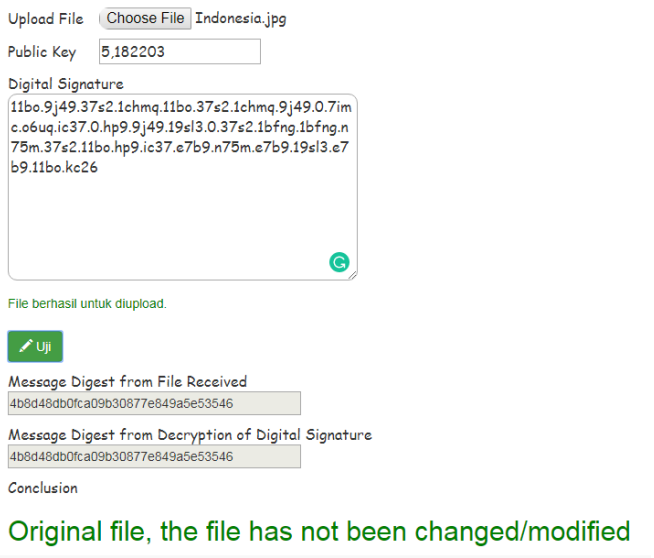
11bo.9j49.37s2.1chmq.11bo.37s2.1chmq.9j49.0.7imc.o6uq.ic37.0.hp9.9j49.19sl3.0.37s2.1bfng.1bfng.n75m.37s2.11bo.hp9.ic37.e7b9.n75m.e7b9.19sl3.e7b9.11bo.kc26

Public key dan *private key* yang dibangkitkan:

Private key : 9a33d,182203
Public key : 5,182203
Processing time : 0.35909104347229 ms.

Setelah *digital signature* dihasilkan, maka *private key* tidak lagi diperlukan. *Public key*, *digital signature*, dan file kemudian dikirim ke penerima. Setelah penerima mendapatkan file dan *public key*, penerima kemudian dapat mendeteksi keaslian file menggunakan *digital signature* dan *public key* tersebut seperti contoh pada gambar berikut:

p-ISSN: 2460-173X
e-ISSN: 2598-5841



Gambar 5. Menguji Keaslian Data Menggunakan Aplikasi yang Dibangun

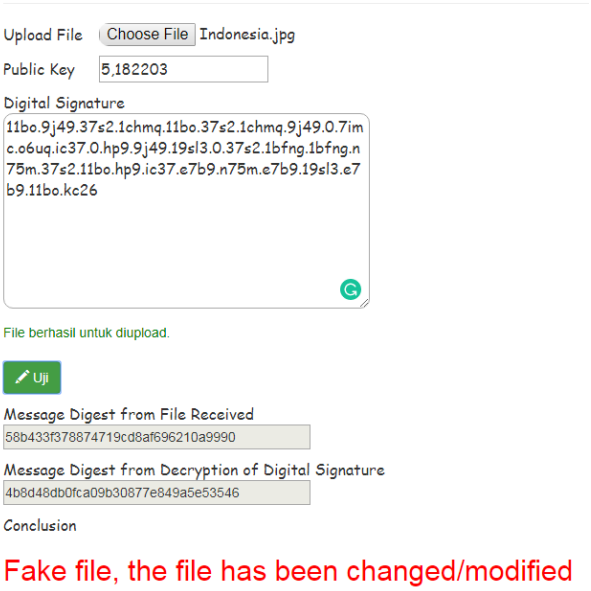
Pada pengujian di atas, dapat dilihat bahwa file Indonesia.jpg terdeteksi asli dan belum diubah/dimodifikasi. Hal ini benar karena file tersebut memang belum dimodifikasi. Proses pemeriksaan dilakukan dengan membandingkan *message digest* dari file yang diterima dengan *message digest* dari dekripsi *digital signature*. Jika keduanya sama atau identik tanpa adanya perbedaan, maka dapat disimpulkan bahwa file tersebut masih asli dan tidak ada perubahan/modifikasi yang terjadi. Namun sebaliknya, jika ada sedikit saja perbedaan walau hanya satu karakter, maka dapat disimpulkan bahwa file tersebut telah dimodifikasi secara ilegal (file palsu).

Pengujian selanjutnya dilakukan dengan melakukan sedikit modifikasi pada file Indonesia.jpg dengan menambahkan titik merah seperti yang ada di lingkaran biru pada gambar berikut:

Gambar 6. File Indonesia.jpg Hasil Modifikasi

Pada gambar di atas dapat dilihat bahwa file Indonesia.jpg telah dimodifikasi dengan menambahkan sebuah titik merah kecil, dimana perubahannya tersebut tidak terlalu terlihat, namun model algoritma yang dirancang harus mampu mendeteksi perubahan sekecil apapun yang terjadi. File hasil modifikasi tersebut diperiksa kembali untuk melihat keakuratan model algoritma yang dirancang. Hasil pengujian dapat dilihat pada gambar berikut:

p-ISSN: 2460-173X
e-ISSN: 2598-5841



Gambar 7. Menguji Keaslian Data dari File Indonesia.jpg yang telah Dimodifikasi

Pada gambar di atas terlihat bahwa file yang terdeteksi mengalami perubahan karena *message digest* dari file yang diterima berbeda dengan *message digest* dari hasil dekripsi *digital signature*. Oleh karena itu, disimpulkan bahwa file tersebut telah diubah atau dimodifikasi secara ilegal (file palsu). Model algoritma yang dirancang dapat mendeteksi perubahan data dengan sangat akurat, meskipun hanya terjadi perubahan kecil yang mungkin tidak terlihat secara kasat mata. Model algoritma dapat membantu penerima menghindari penggunaan file palsu yang merugikan atau membahayakan penerima data.

Siapa saja yang membangkitkan *digital signature* dari file tersebut memerlukan private key. Jika intruder menggunakan sebarang *privat key*, maka *public key* yang digunakan oleh penerima data tidak akan berkorelasi dengan sebarang *privat key* tersebut. Oleh karena itu, untuk memperkuat *digital signature* dan mencegah *private key* digunakan kembali untuk menghasilkan *digital signature* dari file yang sama, maka seyogianya *private key* hanya digunakan sekali dan segera dimusnahkan jika telah digunakan sehingga tidak ada pihak lain yang menggunakannya.

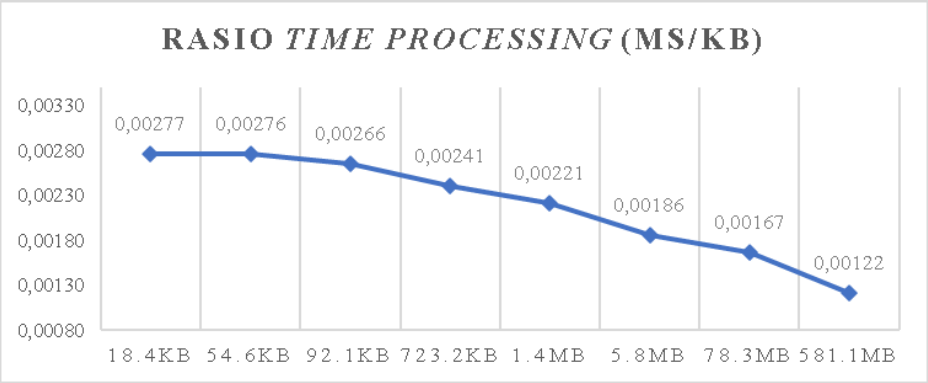
Hasil pengujian untuk berbagai file berukuran dan berekstensi berbeda dapat dilihat pada tabel berikut:

Tabel 1. Hasil Pengujian Terhadap File dengan Ukuran dan Ekstensi File yang Berbeda

No	*File Extension*	*Size*	*Time* (ms)	*Ratio time/size (ms/KB)*
1	.sav	18.4 KB	0.050984	0.00277
2	.docx	54.6 KB	0.150854	0.00276
3	.xlsx	92.1 KB	0.244684	0.00266
4	.pdf	723.2 KB	1.741270	0.00241
5	.ppt	1.4 MB	3.184464	0.00221
6	.mp3	5.8 MB	11.107063	0.00186
7	.mp4	78.3 MB	134.345351	0.00167
8	.mkv	581.1 MB	726.336956	0.00122

Pada tabel tersebut dapat dilihat bahwa algoritma yang diusulkan dapat digunakan untuk menguji keaslian dari berbagai jenis ekstensi file dengan berbagai ukuran. Semakin besar ukuran file maka total waktu proses akan semakin besar, tetapi rata-rata waktu proses per kilo byte semakin menurun. Rasio *time processing* per kilo byte selengkapnya dapat dilihat pada grafik berikut:

p-ISSN: 2460-173X
e-ISSN: 2598-5841



Gambar 8. Rasio *Time Processing* per Kilo Byte

Pada grafik di atas dapat dilihat bahwa semakin besar ukuran file yang diuji, maka waktu proses per kilo byte akan semakin kecil. Hal ini menunjukkan semakin besar ukuran file, maka rata-rata waktu proses per kilo byte akan semakin cepat, sehingga algoritma yang diusulkan mampu membangkitkan *digital signature* dan menguji keaslian file untuk file yang berukuran besar dengan cepat.

Perhitungan MD5 dapat memproses input dengan panjang berapa pun dan akan menghasilkan *message digest* dengan panjang yang tetap yaitu 128bit. Probabiliti untuk mencari *message digest* yang sama untuk *output* 128-bit adalah 2^{128}. Jika diasumsikan seorang intruder dapat melakukan operasi percobaan 1 triliun operasi per detik, maka dibutuhkan waktu hingga $2^{128}/10^{12}$ detik atau 10.790,283 triliun tahun lamanya untuk menemukan pasangan *message digest* yang sesuai. Hal ini menjadi salah satu alasan model rancangan algoritma ini menjadi sangat aman untuk diimplementasikan.

5. KESIMPULAN

Hasil pengujian menunjukkan bahwa *digital signature* yang dihasilkan dari rancangan model dapat memberikan jaminan keamanan terhadap keaslian data. Rancangan ini dapat mencegah penerima data menggunakan data palsu yang telah diubah intruder yang mungkin saja dapat merugikan atau membahayakan penerima data. *Digital signature* yang dihasilkan dari rancangan algoritma juga sangat sensitif terhadap perubahan data, sehingga perubahan satu bit data tetap dapat dideteksi oleh. Hal ini membuat penerima juga dapat mendeteksi keaslian data walau dimodifikasi dengan perubahan yang sangat amat kecil karena perubahan sekecil apapun akan membatalkan *digital signature* yang dihasilkan. Rancangan algoritma pembangkitan *digital signature* pada penelitian ini juga terbukti sangat aman karena memiliki 2^{128} kombinasi *message digest* yang mungkin terjadi. Perhitungan RSA mengantisipasi pihak-pihak yang ingin mengubah *message digest* yang ada atau membuat *digital signature* yang baru dari file hasil modifikasi. Sehingga kombinasi dari MD5 dan algoritma RSA menghasilkan *digital signature* yang kuat dan aman.

REFERENSI

[1] Pooja and M. Yadav, "Digital Signature," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT)*, vol. 3, no. 6, pp. 71-75, 2018.

[2] Y. Yoo, R. Azarderakhsh, A. Jalali, D. Jao, and V. Soukharev, "A Post-quantum Digital Signature Scheme Based on Supersingular Isogenies," in *Financial Cryptography and Data Security*, 2017, pp. 163–181.

[3] L. Ducas *et al.*, "Crystals-dilithium: A Lattice-Based Digital Signature Scheme," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2018, no. 1, pp. 238–268, 2018.

[4] Y. Xue, Y. Tan, C. Liang, Y. Li, J. Zheng, and Q. Zhang, "RootAgency: A Digital Signature-Based Root Privilege Management Agency for Cloud Terminal Devices," *Information Sciences*, vol. 444, pp. 36–50, 2018.
[5] J. H. Seo, "Efficient Digital Signatures from RSA without Random Oracles," *Information Sciences*, vol. 512, pp. 471–480, 2020.
[6] M. Ihwani, "Model Keamanan Informasi Berbasis Digital Signature Dengan Algoritma RSA," *CESS (Journal of Computer Engineering, System and Science)*, vol. 1, no. 1, pp. 15-20, 2016.
[7] R. A. Azdy, "Tanda tangan Digital Menggunakan Algoritme Keccak dan RSA," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, vol. 5, no. 3, pp. 184-191, 2016.
[8] L. Zahhafi and O. Khadir, "A Digital Signature Scheme Based Simultaneously on the DSA and RSA Protocols," *Gulf Journal of Mathematics*, vol. 6, no. 4, pp. 37-43, 2018.
[9] L. B. Sihombing, "Keabsahan Tanda Tangan Elektronik dalam Akta Notaris," *Jurnal Education and Development*, vol. 8, no. 1, pp. 134-134, 2020.
[10] A. Qashlim, "Implementasi Algoritma MD5 untuk Keamanan Dokumen," *Jurnal Ilmiah Ilmu Komputer*, vol. 2, no. 2, 2016.
[11] N. Hayati, M. A. Budiman, and A. Sharif, "Implementasi Algoritma RC4A dan MD5 untuk Menjamin Confidentiality dan Integrity pada File Teks," *SinkrOn*, vol. 1, no. 2, pp. 51–57, 2017.
[12] K. Yusuf, "Penerapan Algoritma MD5 debagai Pengaman Akun pada Aplikasi Web Emusrenbang Kota Binjai," *Jurnal Teknik Informatika Kaputama (JTIK)*, vol. 4, no. 1, pp. 29-34, 2020.
[13] M. M. I. Gultom and D. Saripurna, "Perancangan Sistem Keamanan Aplikasi E-Voting Untuk Pemilihan Ketua Badan Eksekutif Mahasiswa Fakultas Teknik UISU Dengan Menggunakan Algoritma MD5," *Algoritma: Jurnal Ilmu Komputer dan Informatika*, vol. 3, no. 2, pp. 70-77, 2019.
[14] M. R. Rambe, E. V. Haryanto, and A. Setiawan, "Konferensi Nasional Sistem Informasi 2018 STMIK Atma Luhur Pangkalpinang," *IT (Informatic Technique) Journal*, vol.7, no. 1, pp. 51-62, 2018.
[15] S. Suhandinata, R. A. Rizal, D. O. Wijaya, P. Warren, and S. Srinjiwi, "Analisis Performa Kriptografi Hybrid Algoritma Blowfish dan Algoritma RSA," *JURTEKSI (Jurnal Teknologi dan Sistem Informasi)*, vol. 6, no. 1, pp. 1–10, 2019.
[16] Z. Arifin, "Studi Kasus Penggunaan Algoritma RSA sebagai Algoritma Kriptografi yang Aman," *Informatika Mulawarman: Jurnal Ilmiah Ilmu Komputer*, vol. 4, no.3, pp. 7-14, 2016.
[17] Y. Prasetio, B. Triandi, and Hardianto, "Perancangan Aplikasi Pengamanan File Teks dengan Skema Hybrid Menggunakan Algoritma Enigma dan Algoritma RSA," *IT (Informatic Technique) Journal*, vol. 6, no. 1, pp. 46-55, 2018.
[18] E. V. Haryanto, "Desain Steganografi untuk Keamanan Gambar dengan Algoritma RSA dan LSB Berbasis Android," *CSRID (Computer Science Research and Its Development Journal)*, vol. 11, no. 3, p. 179, 2021.

Biodata Penulis

Agung Purnomo Sidik, lahir dan besar di Kota Medan, Sumatera Utara. Menyelesaikan pendidikan Strata-1 jurusan Teknik Informatika di Universitas Pembangunan Panca Budi Medan pada tahun 2014. Menyelesaikan pendidikan Strata-2 jurusan Teknik Informatika di Universitas Sumatera Utara pada tahun 2018. Saat ini aktif mengajar di Universitas Pembangunan Panca Budi Medan dan juga aktif melakukan berbagai penelitian di bidang keamanan komputer dan kecerdasan buatan.