Pemanfaatan *Digital Signature* dalam Menangani Masalah Pencurian Ilustrasi Digital

Ramon Antares Sullivan 18217016
Program Studi Sistem dan Teknologi Informasi/Informatika/Teknik Elektro
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: ramonantares14@gmail.com

***Abstrak*— Ilustrasi digital merupakan salah satu bentuk karya ilustrasi yang sangat sering digunakan saat ini. Ilustrasi digital memberikan kemudahan dalam mencipta serta membagikan karya mereka. Akan tetapi, ilustrasi digital juga sangat rentan terhadap risiko pencurian oleh pihak yang tidak bertanggung jawab. Pemanfaatan tanda tangan digital pada ilustrasi digital diharapkan dapat membantu mengamankan karya ilustrasi digital yang ada.**

Keywords*— *ilustrasi; gambar; digital; tanda tangan; RSA; SHA-1.

I. PENDAHULUAN

Perkembangan teknologi informasi merupakan peristiwa yang memengaruhi aktivitas-aktivitas yang dilakukan oleh manusia dalam kehidupan mereka sehari-hari. Mulai dari pendidikan, ekonomi, dan komunikasi, perkembangan teknologi informasi memberikan dampak terhadap bidang yang mempengaruhi kehidupan masyarakat di Indonesia. Salah satunya adalah bidang ilustrasi. Ilustrasi digital merupakan medium yang umum digunakan saat ini sebagai media ekspresi seni. Ilustrasi digital membantu orang-orang dari berbagai kalangan dan latar belakang dengan memberikan kemudahan untuk mereka dalam menciptakan karya melalui media digital. Hal ini menyebabkan perkembangan industri kreatif di Indonesia menjadi lebih pesat.

Akan tetapi, dengan munculnya teknologi baru, tantangan dan masalah baru juga muncul. Di dalam bidang ilustrasi dan industri kreatif di Indonesia, masalah pencurian karya masih marak terjadi. Ditambah lagi, ilustrasi digital menjadi salah satu target karya yang berpotensi besar untuk dicuri. Pencipta karya ilustrasi digital dapat dengan mudah mengunggah karya mereka ke media sosial maupun *platform* khusus tempat mereka membagikan karya mereka. Namun, karya mereka juga dapat dengan mudah dicuri oleh pihak-pihak yang tidak bertanggung jawab dan digunakan dengan sembarangan oleh mereka. Pencurian ilustrasi digital paling rentan terjadi pada ilustrator nonprofesional, di mana mereka tidak memiliki hak intelektual terhadap karya mereka dan membuat karya sebagai hobi dan mengunggah karya mereka ke internet supaya dapat dilihat oleh orang lain. Mereka tidak memiliki perlindungan hukum karena karya mereka tidak memiliki hak cipta yang diakui secara hukum.

Meskipun begitu, mereka masih memiliki hak terhadap karya yang mereka ciptakan dan membutuhkan cara untuk memberikan kemanan terhadap karya mereka. Salah satu cara yang dilakukan oleh ilustrator adalah memberikan watermark secara manual pada karya mereka seperti nama atau tanda tangan dalam karya mereka. Namun, metode ini masih belum dapat menjamin keamanan dari ilustrasi mereka karena pencuri ilustrasi dapat dengan mudah menghapus watermark maupun tanda tangan mereka. Oleh karena itu, salah satu solusi yang mungkin dalam menangani permasalahan pencurian ilustrasi digital adalah dengan menggunakan *digital signature* pada karya mereka.

II. DASAR TEORI

A. RSA

RSA adalah sebuah algoritma kunci publik yang dibuat oleh Ron Rivest, Adi Shamir, dan Leonard Adleman pada tahun 1976. RSA merupakan singkatan dari nama belakang dari ketiga penemunya, yaitu Rivest, Shamir, dan Adleman. RSA menggunakan kunci publik untuk mengenkripsi dan privat untuk mendekripsi pesan.

Algoritma RSA memiliki 3 proses utama, yaitu pembangkitan kunci publik dan privat, enkripsi, dan dekripsi. Pada proses pertama, algoritma RSA akan membangkitkan sepasang kunci publik dan kunci privat. Proses pembangkitan tersebut meliputi.

1. Menentukan 2 bilangan prima p dan q.
2. Menghitung nilai modulus n, n merupakan nilai dari
$$n = pq$$
3. Menghitung fungsi *totient* $\phi(n)$. Fungsi $\phi(n)$ menghitung bilangan yang relatif prima terhadap p dan q.
$$\phi(n) = (p - 1)(q - 1)$$
4. Menentukan sebuah bilangan bulat e sebagai kunci publik. Nilai e harus relatif prima terhadap $\phi(n)$.
5. Menghitung kunci dekripsi d. Nilai d dihitung dengan persamaan
$$ed \equiv 1 \pmod{\phi(n)} \text{ atau } d \equiv e^{-1} \bmod (\phi(n)).$$

Hasil dari algoritma di atas merupakan sepasang kunci publik (e,n) dan kunci privat (d,n) [1].

roses selanjutnya adalah proses enkripsi. Proses enkripsi dari algoritma RSA meliputi [1].

1. Menyatakan pesan menjadi blok-blok plainteks: m_1, m_2, m_3 , … m_i dengan syarat

$$0 \leq m_i < n - 1.$$

2. Menghitung blok *ciphertext* c_i untuk blok plainteks m_i menggunakan kunci publik e dengan persamaan

$$c_i = m_i \, e \bmod n.$$

Setelah enkripsi, proses selanjutnya dalam RSA adalah proses dekripsi. Misalkan diberikan blok-blok *ciphertext* berupa c_1, c2, c3, … ci. Blok plainteks mi akan dihitung kembali dari blok cipherteks ci dengan menggunakan kunci privat d dengan persamaan

$$m_i = c_i \, d \bmod n.$$

Algoritma RSA memiliki keamanan yang cukup tinggi yang berasal dari tingkat kesulitan dalam memfaktorkan bilangan bulat n faktor-faktor prima bilangan p dan q [1].

B. SHA-1

SHA (*Secure Hash Algorithm*) merupakan fungsi hash searah yang dibuat oleh NIST dan digunakan bersama DSS (Digital Signature Standard). SHA merupakan standar dari fungsi hash searah. SHA dibuat berdasarkan MD4 yang dibuat oleh Ronald L. Rivest dari MIT [2].

Algoritma SHA menerima masukan yang memiliki ukuran maksimum 2^{64} bit dan menghasilkan message digest yang memiliki panjang 160 bit. Dibandingkan dengan MD5 yang memiliki message digest sepanjang 128 bit, message digest yang dihasilkan oleh SHA lebih panjang [2].

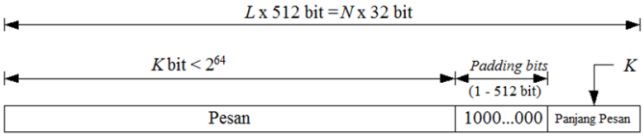
SHA merupakan sebutan umum untuk keluarga fungsi hash satu-arah. SHA memiliki 6 varian [2]:

1. SHA-0
2. SHA-1
3. SHA-224
4. SHA-256
5. SHA-384
6. SHA-512

Pembuatan message digest menggunakan algoritma SHA dibagi menjadi 4 langkah [2]:

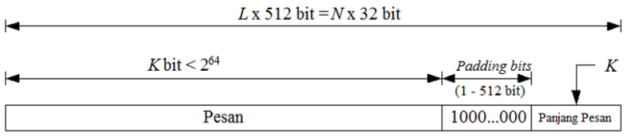
1. Menambahkan bit-bit pengganjal (*padding bits*).
2. Menambahkan nilai panjang pesan.
3. Menginisialisasi penyangga (*buffer*) MD.
4. Mengolah pesan dalam blok berukuran 512 bit.

Pada langkah pertama, pesan akan ditambah dengan sejumlah bit pengganjal (*padding bits*) sedemikian sehingga panjang pesan (dalam satuan bit) kongruen dengan 448 (mod 512). Bit-bit pengganjal memiliki panjang antara 1 sampai 512 bit. Bit-bit pengganjal terdiri dari sebuah bit pertama yang bernilai 1, lalu diikuti bit 0 sebagai nilai bit seterusnya [2].



Gambar II.1 Ilustrasi penambahan bit-bit pengganjal (Sumber: Slide Kuliah *Secure Hash Algorithm (SHA)*)

Selanjutnya, pesan yang telah diberi bit-bit pengganjal akan ditambah lagi dengan 64 bit yang menyatakan panjang pesan semula [2].



Gambar II.2 Ilustrasi penambahan nilai panjang pesan (Sumber: Slide Kuliah *Secure Hash Algorithm (SHA)*)

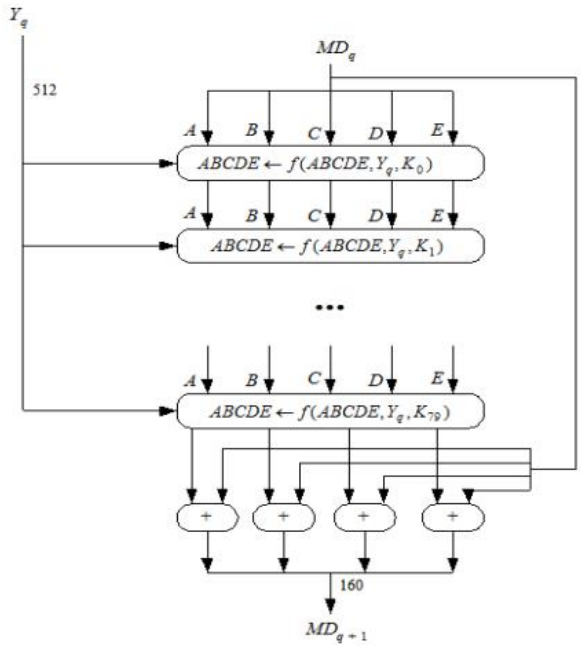
Selanjutnya adalah proses inisialisasi penyangga MD. Penyangga MD merupakan 5 buah bit-bit penyangga (*buffer*) yang masing-masing panjangnya 32 bit dengan total panjang penyangga sebesar $5 \times 32 = 160$ bit [2]. Kelima penyangga MD ini diberi nama A, B, C, D, dan E. Setiap penyangga MD diinisialisasi dengan nilai-nilai dalam notasi HEX sebagai berikut [2]:

1. A = 67452301,
2. B = EFCDAB89,
3. C = 98BADCFE,
4. D = 10325476,
5. E = C3D2E1F0.

Proses pengolahan pesan H_{SHA} terdiri dari 80 buah putaran. Masing-masing putaran menggunakan bilangan penambah K_t yang memiliki nilai yang berbeda pada interval waktu t tertentu [2].

1. Putaran $0 \leq t \leq 19$ Kt = 5A827999
2. Putaran $20 \leq t \leq 39$ Kt = 6ED9EBA1
3. Putaran $40 \leq t \leq 59$ Kt = 8F1BBCDC
4. Putaran $60 \leq t \leq 79$ Kt = CA62C1D6.

Pengolahan pesan dalam blok berukuran 512 bit dapat digambarkan melalui diagram di bawah ini [2].

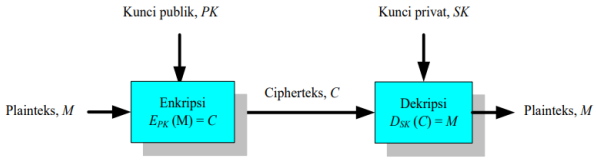


Gambar II.3 Ilustrasi pengolahan pesan dalam blok 512 bit (Sumber: Slide Kuliah *Secure Hash Algorithm (SHA)*)

C. Tanda Tangan Digital

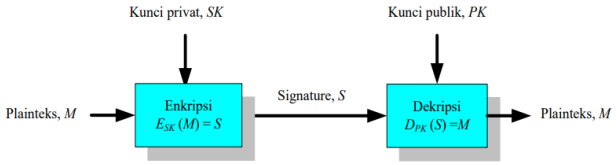
Tanda tangan digital adalah salah satu jenis dari tanda tangan elektronik yang menggunakan algoritma matematis untuk melakukan validasi terhadap keaslian dan integritas suatu pesan [3]. Tanda tangan digital digunakan sebagai otentikasi dan identifikasi pengguna serta melindungi informasi dalam pesan atau dokumen digital dengan memastikan tidak ada perubahan yang terjadi selama transmisi antara pengirim pesan dan penerima.

Tanda tangan digital dilakukan menggunakan kriptografi kunci-publik. Kriptografi kunci publik adalah metode kriptografi yang menggunakan sistem pasangan kunci privat dan kunci publik [3]. Proses enkripsi menggunakan kriptografi kunci publik dilakukan dengan mengenkripsi pesan dengan kunci publik penerima, lalu *ciphertext* akan dikirim ke penerima dan didekripsi menggunakan kunci privat milik penerima [4].



Gambar II.4 Ilustrasi enkripsi kunci publik (Sumber: Slide Kuliah Tanda Tangan Digital)

Proses otentikasi dilakukan dengan menggunakan kunci privat pengirim. Pesan akan dienkripsi menggunakan kunci privat pengirim, yang dapat didekripsi menggunakan kunci publik pengirim oleh penerima pesan. Kerahasiaan dan otentikasi pesan dapat dicapai dengan cara ini [4].



Gambar II.5 Ilustrasi enkripsi tanda tangan digital (Sumber: Slide Kuliah Tanda Tangan Digital)

Namun, kedua pihak harus memiliki sertifikat digital yang terdaftar dari otoritas yang bertanggung jawab untuk menghubungkan penandatangan dan tanda tangan mereka. Enkripsi adalah proses penyandian data yang dikirim ke penerima dalam bentuk yang hanya dapat didekripsi oleh penerima. Otentikasi adalah proses validasi informasi dari pengirim yang asli dan belum diubah dalam perjalanan [3].

D. Gambar Digital

Gambar digital adalah representasi dari gambar asli sebagai sekelompok angka yang dapat disimpan dan diproses oleh komputer. Untuk mengubah gambar menjadi angka, gambar dibagi menjadi area kecil yang disebut piksel (elemen gambar). Untuk setiap piksel, perangkat pencitraan merekam sebuah angka, atau sekumpulan kecil angka, yang menjelaskan beberapa properti dari piksel tersebut, seperti kecerahannya (intensitas cahaya) atau warnanya. Angka-angka tersebut disusun dalam sebuah *array* baris dan kolom yang sesuai dengan posisi vertikal dan horizontal piksel pada gambar [5].

Gambar digital memiliki beberapa karakteristik. Pertama adalah tipe gambar. Sebagai contoh, gambar hitam-putih hanya menyimpan intensitas cahaya yang ada pada setiap pikselnya. Gambar berwarna dapat direpresentasikan melalui ruang warna, seperti RGB atau CYMK. Selain itu, gambar digital juga memiliki resolusi tertentu. Resolusi suatu gambar digital diukur menggunakan satuan ppi (pixel per inch). Gambar dengan resolusi yang tinggi akan menghasilkan gambar yang lebih detail [5].

Dalam gambar digital berwarna, kedalaman warna dari suatu gambar direpresentasikan dalam bits per pixel. Semakin banyak bits dari suatu gambar digital, maka semakin banyak juga nilai kedalaman dan jenis warna yang dapat disimpan dalam gambar. Gambar digital biasanya disimpan dalam beberapa format, seperti JPEG, GIF, PNG, dan TIFF [5].

Salah satu keuntungan dari gambar digital diibandingkan gambar tradisional adalah kemampuan untuk mengirim gambar secara elektronik dengan mudah dan cepat ke medium yang berbeda. Selain itu, gambar digital juga dapat diubah dengan mudah sesuai dengan keinginan pengguna menggunakan aplikasi tertentu, seperti Photoshop atau Gimp [5].

III. ANALISIS DAN RANCANGAN

Pemanfaatan tanda tangan digital dapat digunakan dalam ilustrasi digital sebagai bentuk otentikasi terhadap karya ilustrasi digital. Makalah ini akan menjelaskan proses implementasi dari tanda tangan digital dalam ilustrasi digital. Tanda tangan digital akan diimplementasi menggunakan algoritma RSA dan SHA-1.

A. *Skema Tanda Tangan Digital Sebagai Bentuk Otentikasi Terhadap Karya Ilustrasi Digital*

Dalam skema ini, diperlukan sebuah program yang dapat membuat tanda tangan digital pada karya ilustrasi digital dengan format gambar digital. Program dapat membangkitkan sepasang kunci publik dan privat yang berfungsi dalam memberikan tanda tangan digital dalam *file* gambar. *File* yang sudah ditandatangani menggunakan kedua kunci tadi selanjutnya perlu diverifikasi untuk memastikan bahwa tanda tangan digital tersebut dapat menjamin otentikasi dari *file* gambar tersebut.

IV. IMPLEMENTASI

Bagian ini akan melakukan proses implementasi terhadap skema yang telah dirancang. Program dibuat dengan bahasa pemrograman Python. Library yang digunakan adalah electron.js untuk keperluan antarmuka program. Selain itu, program juga menggunakan library bawaan Python untuk fungsi SHA-1. Algoritma yang digunakan adalah algoritma RSA dengan menggunakan fungsi *hash* SHA-1.

Program berisi menu pembangkitan kunci publik dan privat RSA, menu pembangkitan tanda tangan digital (*file signing*), dan menu verifikasi tanda tangan digital (*verifying*). Program menerima masukan berupa *file* gambar digital. *File* gambar digital yang akan digunakan dalam implementasi ini bertipe JPEG. Program akan membangkitkan kunci publik dan privat RSA yang akan digunakan dalam proses penandatanganan digital *file* gambar. Setelah *file* gambar diberi tanda tangan digital, program juga dapat melakukan verifikasi terhadap *file* tersebut menggunakan kunci privat yang dibangkitkan. Beberapa dokumentasi program yang digunakan dalam makalah ini dapat dilihat pada bagian-bagian di bawah ini.

```
def generate_keypair(byte_size = 16):
    a, b =
get_random_big_prime_pair(byte_size //
2)
    n = a * b
    fn = (a - 1) * (b - 1)

    e = get_relative_prime(fn)
    d = get_congruen(e, 1, fn)

    nb = get_base_64(n, byte_size)
    eb = get_base_64(e, byte_size)
    db = get_base_64(d, byte_size)
    pubkey = f'{eb}:{nb}'
    privkey = f'{db}:{nb}'

    return pubkey, privkey
```

Program IV.1 *source code* generate_keypair.py (Sumber: dokumentasi penulis)

Program di atas merupakan program untuk membangkitkan sepasang kunci publik dan privat RSA. Kedua kunci ini digunakan dalam tanda tangan digital pada *file* gambar digital.

```
def encrypt_message(text: bytes, key:
int, n: int) -> bytes:
    byte_size = (n.bit_length() + 7) //
8
    cipher_ints = []
    cipher_bytes = []

    for i in range(0, len(text),
byte_size):
        current_bytes = text[i:i +
byte_size]
        current_int =
int.from_bytes(current_bytes, 'little')

        cipher_int = pow(current_int,
key, n)
        cipher_byte =
cipher_int.to_bytes(byte_size, 'little')

        cipher_bytes += cipher_byte

    return bytes(cipher_bytes)
```

Program IV.2 *source code* encrypt_message.py (Sumber: dokumentasi penulis)

Program di atas merupakan program untuk enkripsi *file* yang akan diberikan tanda tangan digital.

```
if __name__ == '__main__':
    file_path = sys.argv[1]
    is_save_sign = sys.argv[2]
    pubkey = sys.argv[3]

    with open(file_path, 'rb') as
file_buf:
        target_file = file_buf.read()

    file_hash =
hashlib.sha1(target_file).hexdigest()
    file_hash =
file_hash.encode('ascii')

    file_signature = encrypt(file_hash,
pubkey)
    print(file_signature.hex())

    if is_save_sign == 'true':
        with open(file_path, 'a+b') as
file_buf:
            file_buf.write(

f'<ds>{file_signature.hex()}</ds>'.encod
e('ascii')
            )
```

Program IV.3 *source code* sign_*file*.py (Sumber: dokumentasi penulis)

Program di atas merupakan program untuk menandatangani *file* gambar menggunakan tanda tangan digital.

```
if __name__ == '__main__':
    file_path = sys.argv[1]
    is_save_sign = sys.argv[2]
    privkey = sys.argv[3]
    signature = sys.argv[4]

    with open(file_path, 'rb') as
file_buf:
        target_file = file_buf.read()

    if is_save_sign == 'true':
        signature = target_file[-101:-
5]
        signature = signature.decode()
        target_file = target_file[:-
105]

    signature =
bytes.fromhex(signature)
    decrypted_signature =
decrypt(signature, privkey)
    decrypted_signature =
decrypted_signature.decode('ascii')
    decrypted_signature =
decrypted_signature.replace('\x00', '')
    file_hash =
hashlib.sha1(target_file).hexdigest()

    if decrypted_signature ==
file_hash:
        print('File has valid
signature')
    else:
        print('Invalid signature')
```

Program IV.4 *source code* program verify_*file*.py (Sumber: dokumentasi penulis)

Program di atas merupakan bagian program yang berfungsi untuk melakukan verifikasi terhadap tanda tangan digital yang ada pada *file* gambar.

V. Hasil Uji Coba

Uji coba dari skema program akan dilakukan menggunakan sebuah *file* gambar digital ilustrasi. *File* ilustrasi berupa *file* dengan tipe JPEG.

Gambar V.1 Ilustrasi digital awal (Sumber: dokumentasi penulis)

Program akan membangkitkan sepasang kunci publik dan privat yang akan digunakan dalam proses tanda tangan digital pada *file* gambar ilustrasi digital tersebut. Kunci publik yang digunakan dalam tanda tangan digital ini adalah "X0+Fm8YB7Hs0MO2ELB51CA==:r3Wa5FKAQx6wa0EsInm6oA==", sedangkan kunci privat yang digunakan adalah "H0mstEEy2Kh7ynsNUUnsRw==:r3Wa5FKAQx6wa0EsInm6oA==".

Setelah memperoleh sepasang kunci publik dan privat untuk tanda tangan digital, tahap selanjutnya adalah tahap *signing*. Di dalam tahap ini, *file* gambar akan diberikan tanda tangan digital berupa *file signature* di dalam *binary file* gambar tersebut. Format *file signature* di dalam *file* gambar digitalnya adalah <ds>623b5a4d27a1d79c92309226cd8c4b11b9aef6d2a0be29fef23370e08b68bf107e088fc6f53e35f8988b54e98f680515</ds> dipisahkan menggunakan tag <ds> </ds>.

Selanjutnya, beberapa pengujian akan dilakukan terhadap *file* gambar yang sudah memiliki tanda tangan digital untuk menguji otentikasi gambar yang sudah diberikan tanda tangan digital. Pengujian dilakukan dengan melakukan verifikasi terhadap *file* gambar ketika gambar disunting, mengubah nilai *file signature* dalam *file*, dan menggunakan kunci privat lain untuk melakukan verifikasi terhadap *file*.

Kasus Pengujian	Hasil
Verifikasi setelah gambar disunting.	Invalid signature. Program dapat mendeteksi perubahan pada *file* gambar.
Verifikasi setelah nilai *file signature* diubah.	Invalid signature. Program dapat mendeteksi perubahan pada nilai *file signature* gambar.
Verifikasi *file* menggunakan kunci privat yang berbeda.	Invalid signature. Program dapat mendeteksi penggunaan kunci privat yang berbeda.

Tabel V.1 Tabel hasil pengujian program tanda tangan digital.

VI. KESIMPULAN

Pemanfaatan tanda tangan digital dalam ilustrasi digital dapat dijadikan salah satu alternatif bagi pembuat karya ilustrasi digital dalam melindungi karya mereka dari pencurian. Tanda tangan digital memiliki beberapa kelebihan yang dapat menjaga otentikasi dari kepemilikan ilustrasi digital.

Pertama, tanda tangan digital dapat menjamin otentikasi dari ilustrasi digital yang dapat diverifikasi dengan mudah. Tanda tangan digital tidak secara kasat mata muncul di dalam gambar, melainkan tertanam di dalam *binary file* dari gambar tersebut. Hal ini dapat membantu para pembuat karya ilustrasi sehingga pencuri karya ilustrasi tidak dapat dengan mudah langsung mengetahui bahwa adanya tanda tangan digital yang terenkripsi di dalam file tersebut.

Selanjutnya, tanda tangan digital tidak mengubah kualitas gambar digital yang dienkripsi. Pembuat karya ilustrasi tidak perlu khawatir jika kualitas karya ilustrasi mereka akan berubah karena tanda tangan digital karena tanda tangan digital tidak mengubah isi dari *binary file* gambar digital, sehingga kualitas gambar terjaga. Penggunaannya juga tergolong mudah karena kedua kunci privat dan publik yang digunakan dibangkitkan secara otomatis. Selain dari sisi pengguna pembuat karya ilustrasi digital, algoritma RSA dan SHA-1 juga tergolong aman digunakan sebagai tanda tangan digital. Untuk ke depannya, studi kasus mengenai pemanfaatan tanda tangan digital pada ilustrasi digital perlu diteliti lebih jauh aplikasinya, terutama pada kasus-kasus tertentu seperti gambar yang disunting dengan suntingan yang banyak.

REFERENSI

[1] Munir, R. 2021. Slide Kuliah Algoritma RSA.

[2] Munir, R. 2018. Slide Kuliah *Secure Hash Algorithm (SHA)*.

[3] Paul, E. 2017. "What is Digital Signature – How it works, Benefits, Objectives, Concept". URL: https://www.emptrust.com/blog/benefits-of-using-digital-signatures. EMP Trust HR. Diakses pada 20 Mei 2021.

[4] Munir, R. 2021. Slide Kuliah Tanda Tangan Digital.

[5] 2021. “Digital Images”. URL: https://www.encyclopedia.com/computing/news-wires-white-papers-and-books/digital-images. Diakses pada 20 Mei 2021.

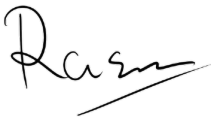
VIDEO LINK AT YOUTUBE

https://youtu.be/XQJhqUD4zPs .

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 26 April 2021



Ttd

Ramon Antares Sullivan 18217016