Implementasi Tanda Tangan Digital Menggunakan Kombinasi Kriptografi Kunci Publik dan Fungsi Hash, serta Audio Steganografi pada Karya Musik Instrumental

Harith Fakhiri Setiawan - 13519161
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13519161@std.stei.itb.ac.id

***Abstraksi*—Terpublikasinya berbagai karya digital berupa musik instrumental tanpa adanya label rekaman yang menjaga hak paten dan hak cipta semakin marak seiring berkembangnya era digitalisasi. Hal ini membuat seringnya ditemukan permasalahan terkait pengambilan karya digital musik instrumental oleh pihak-pihak tidak bertanggung yang menggunakannya untuk kepentingan pribadi. Oleh karena itu, diperlukan solusi untuk mengatasi pengambilan atau perebutan hak paten serta hak cipta bagi para musisi kecil berbakat yang belum memiliki label rekaman dan belum dikenal masyarakat. Solusi ini diterapkan dengan menggunakan implementasi tanda-tangan digital menggunakan kombinasi kriptografi kunci publik dan fungsi *hash*, serta audio steganografi. Hasil penelitian menunjukan bahwa solusi yang ditawarkan dapat menyelesaikan permasalahan yang diangkat pada topik makalah ini.**

Keywords—hash; tanda-tangan digital; RSA; SHA-256; kunci publik; message-digest

I. Pendahuluan

Seiring berlangsungnya digitalisasi, pembuatan karya yang dipublikasi pada ranah digital adalah hal yang semakin marak. Berbagai jenis karya, salah satunya karya karya musik instrumental. Terdapat banyak musisi kecil berbakat yang berlomba-lomba dalam menciptakan karyanya masing-masing walau belum dikenal banyak orang, atau belum memiliki kontrak dengan sebuah label rekaman. Musisi ini mempublikasi karya musik instrumen ciptaan mereka dengan berbagai tujuan, mulai dari sekedar menyalurkan hobi, hingga menunggu ada label rekaman yang menawarkan kontrak dengan mereka.

Tanpa disadari, hal ini juga membuka celah baru bagi pihak-pihak yang tidak bertanggung jawab untuk dapat mengambil karya cipta orang lain dengan lebih mudah. Terpublikasi berbagai karya musik tanpa label pada platform yang tidak menjamin perlindungan hak cipta ataupun yang biasa dikenal dengan istilah *copyright* inilah yang menjadi alasan mengapa sekarang ini sering terjadi permasalahan perebutan hak cipta serta *copyright* untuk karya musik digital, terutama bagi para musisi kelas bawah yang masih belum menjalin kontrak dengan label yang dapat membantu dalam penjagaan hak cipta.

Oleh karena itu, pada makalah ini akan dibahas implementasi tanda-tangan digital menggunakan kombinasi kriptografi kunci publik dan fungsi *hash*, serta audio steganografi pada karya musik instrumental. Dengan implementasi ini, diharapkan dapat membantu terjaganya hak cipta dan hak paten bagi musisi kecil berbakat, sehingga dapat meminimalisasi terjadinya perebutan hak cipta dan hak paten pada karya musik instrumental.

II. Dasar Teori

A. *Digital Signature*

Digital Signature atau tanda-tangan digital adalah sebuah nilai kriptografis yang dibuat sesuai dengan isi pesan dan kuncinya, dimana setiap *sign* atau tanda tangan yang diberikan selalu berbeda dan bersifat untuk antar satu pesan dengan pesan lainnya, dan/atau antara satu kunci dengan kunci lainnya. Tanda-tangan digital memberikan dua dari 4 layanan kriptografi, yaitu otentikasi (*authentication)* dan anti-penyangkalan (*nonrepudiation*).[1] Tanda-tangan digital memiliki karakteristik yang sama dengan tanda-tangan non digital. Perbedaannya hanya terletak pada tanda-tangan non digital yang hanya unik untuk setiap orang, sedangkan tanda-tangan digital yang selalu unik untuk setiap orang dan dokumennya.[1]

Secara garis besar, terdapat dua proses dalam tanda-tangan digital, yaitu menandatangani pesan (*signing*), dan memverifikasi pesan (*verification*). Pada proses penandatangan pesan, terdapat dua metode yang umum dilakukan, yaitu dengan mengenkripsi pesan untuk pesan rahasia, serta kombinasi fungsi hash dan kriptografi

B. Fungsi *Hash* SHA-256

Fungsi hash adalah sebuah fungsi yang bertujuan untuk mengkompresi pesan (*M*) yang berukuran sembarang menjadi string (*h*) yang berukuran tetap. Luaran dari fungsi *hash* adalah sebuah pesan ringkas atau biasa dikenal dengan *message-digest*. Fungsi *hash* bersifat *irreversible*, yang artinya *message-digest* yang dihasilkan tidak dapat dikembalikan menjadi pesan semula.

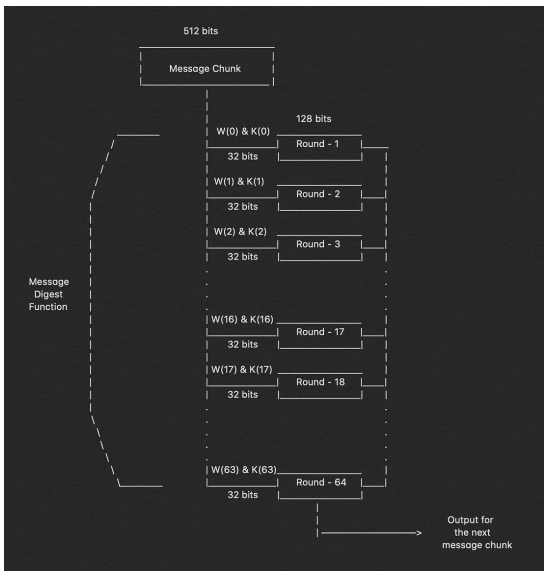
Berikut adalah persamaan fungsi *hash*.[3]

$$h = H(M)$$

SHA (*Secure Hash Algorithm*) adalah adalah fungsi *hash* satu-arah yang dibuat oleh *NIST* dan digunakan bersama *DSS* (*Digital Signature Standard*). Salah satu fungsi *hash* SHA yang cukup umum digunakan dalam digital signature adalah fungsi *hash* SHA-256 dan akan menjadi fungsi *hash* yang digunakan pada makalah ini.

Secara garis besar, langkah-langkah pengerjaan SHA-256 adalah sebagai berikut.

1. Tambahkan bit pada masukan *hash* sehingga panjang bit adalah tepat kurang 64 dari kelipatan 512.
2. Tambahan bit panjang atau *length bits* yang berisikan tepat 64 bit agar pesan masukan menjadi tepat kelipatan dari 512.
3. Inisiasi buffer 8 *buffer* yang akan digunakan untuk melakukan kompresi.
4. Pesan dengan panjang kelipatan dari 512 (*n x 512*) akan dibagikan menjadi n buah *message chunk* yang masing-masing berukuran 512 bit untuk dijalankan pada 64 putaran operasi.
5. Luaran yang dihasilkan pada setiap putaran akan menjadi masukan pada putaran selanjutnya, dan dilakukan hingga putaran terakhir pada *message chunk* ke *n*. Luaran terakhir memiliki ukuran 256 bit.[2]



Gambar **1.** skema operasi pada setiap *message chunk*[2].

C. Kriptografi Kunci Publik RSA

Algoritma RSA (Rivest-Shamir-Adleman) adalah sebuah algoritma kunci-publik yang terkenal dan memiliki banyak aplikasi. Keamanan algoritma RSA bertitik tumpu pada kesulitan pemfaktoran bilangan bulat besar yang menjadi faktor-faktor prima. Properti dari algoritma antara lain:

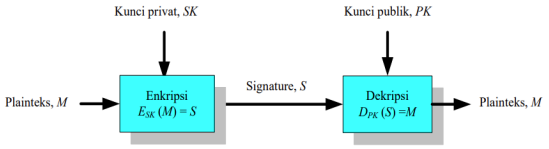
1. *p* dan *q* yang merupakan bilangan prima (rahasia)
2. $n = p.q$ (tidak rahasia)
3. $\phi(n) = (p-1)(q-1)$ (rahasia)
4. *e* (kunci enkripsi) (tidak rahasia)
5. *d* (kunci dekripsi) (rahasia)
6. m (plainteks) (rahasia)
7. c (cipherteks) (tidak rahasia).

Algoritma ini memiliki dua persamaan umum yang digunakan untuk melakukan enkripsi dan juga dekripsi sebagai berikut.[4]

$$\text{Enkripsi} : E_e(m) = m^e \bmod n = c$$

$$\text{Dekripsi} : D_d(m) = c^d \bmod n = m$$

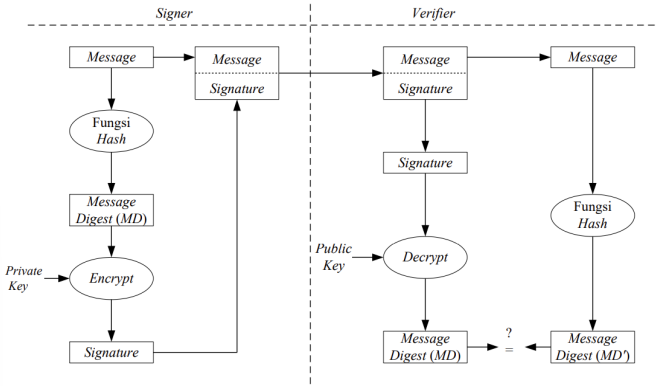
Namun, pada aplikasi tanda-tangan digital menggunakan kriptografi kunci-publik dan fungsi hash, terdapat perbedaan dimana kunci enkripsi bersifat rahasia menggunakan kunci privat, sementara kunci dekripsi bersifat rahasia menggunakan kunci publik. Hal ini bertujuan untuk menyimpan kerahasiaan dari pesan yang dikirimkan oleh pengirim. Secara garis besar, skema aplikasi penandatanganan digital menggunakan kriptografi kunci-publik adalah sebagai berikut.



Gambar **2.** skema kriptografi kunci publik dengan RSA[1]

D. Aplikasi Penandatangan dengan Menggunakan Kombinasi Kriptografi kunci-publik dan Fungsi Hash

Kriptografi kunci-publik dan fungsi hash dapat digunakan dikombinasikan dan digunakan untuk proses tanda-tangan digital. Kombinasi ini dikenal dengan proses penandatanganan dengan kombinasi kriptografi kunci-publik dan fungsi hash. Secara garis besar, skema penandatangan kombinasi ini adalah sebagai berikut.[1]



Gambar **3.** skema penandatangan dengan menggunakan kombinasi kriptografi kunci-publik dan fungsi *hash*[1]

E. Steganografi

Steganografi adalah seni menyembunyikan pesan atau informasi rahasia dalam suatu media tanpa menarik perhatian orang yang tidak berwenang. Teknik ini telah digunakan sejak zaman kuno untuk tujuan menyampaikan pesan rahasia. Selain

hanya menyampaikan pesan rahasia, steganografi juga dapat diterapkan pada masa kini untuk penandaan hak cipta, hak paten, dan lain-lain.

Sekarang ini, penggunaan steganografi dapat diaplikasikan pada berbagai media digital, diantaranya gambar digital, audio, video, serta dokumen teks. Steganografi pada audio digital adalah steganografi yang akan dibahas pada makalah ini.

III. Rancangan Solusi dan Implementasi

Dalam rangka penyelesaian permasalahan, perlu dilakukan beberapa langkah untuk dapat menguasai serta menentukan solusi yang tepat terkait topik permasalahan yang diangkat. Langkah-langkah tersebut adalah sebagai berikut.

A. Gambaran Umum Solusi dari permasalahan

Untuk menyelesaikan permasalahan dari topik yang diangkat, yaitu pencegahan pengambilan hak milik sebuah karya instrumental tanpa sepengetahuan pencipta, dilakukan pendekatan solusi menggunakan metode penandatangan dengan menggunakan kombinasi kriptografi kunci-publik dan fungsi hash. Kriptografi kunci-publik yang digunakan adalah algoritma RSA dan Fungsi *hash* yang dipilih adalah fungsi *hash* SHA-256.

Alasan dipilihnya algoritma RSA sebagai salah satu pendekatan solusi adalah algoritma RSA yang menggunakan *asymmetric key*, yang terbagi menjadi kunci privat dan kunci publik, yang dapat meningkatkan keamanan dari penyerangan. Selain itu, algoritma RSA yang memanfaatkan faktor bilangan prima dari bilangan yang besar juga mempersulit pihak luar untuk memecahkan kunci dari RSA dan dapat meningkatkan keamanan.

Sementara itu, fungsi *hash* SHA-256 dipilih karena fungsi hash ini memiliki tingkat keamanan yang sangat tinggi, dimana luaran memiliki ukuran 256 bit, yang berarti terdapat 2^{256} kemungkinan luaran yang dapat dihasilkan. Hal ini menyebabkan sangat kecil kemungkinan hasil *message-digest* yang diciptakan dapat dicari ataupun memiliki nilai lain yang dapat menghasilkan *message-digest* yang sama.

B. Rancangan Umum Solusi Permasalahan

Rancangan solusi dari permasalahan yang dipilih pada topik ini secara garis besar dibagi menjadi 2 pihak inti, dan 1 pihak opsional. Pihak inti terdiri dari pencipta karya musik instrumental yang mempublikasikan karya musik instrumen nya dan pihak penyedia platform yang publikasi, sementara pihak opsional adalah pihak yang mengaku-ngaku sebagai pemilik karya musik instrumental. Serangan pihak opsional tidak akan dibahas secara detail pada makalah ini. Proses dilakukan dengan melakukan penanaman tanda-tangan digital sampai dengan verifikasi tanda-tangan digital. =

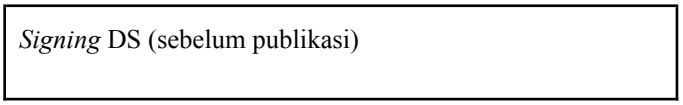
Proses dimulai dengan pihak inti 1, yaitu pencipta karya musik instrumental sebelum melakukan publikasi terhadap karya musik instrumental yang belum bernaungan di bawah sebuah label pada sebuah platform publikasi sebagai pihak inti 2.

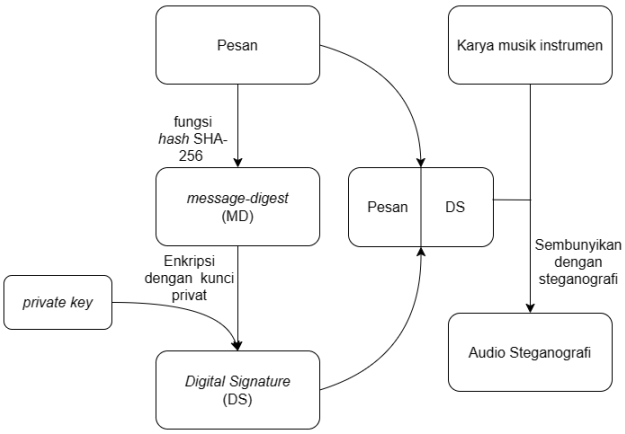
1. Pihak inti 1 membuat sebuah pasangan dengan algoritma RSA.
2. Pihak inti 1 membuat sebuah pesan, yang menandakan bahwa lagu tersebut adalah hak milik pihak inti 1. Contoh pesan adalah “lagu instrumen piano ini adalah ciptaan harith fakhiri”
3. Pesan tersebut kemudian di-*hash* menjadi sebuah *message-digest*, kemudian dienkripsi menggunakan kunci privat yang dibuat sebelumnya.
4. Hasil dari enkripsi tersebut adalah *digital signature* atau tanda-tangan digital. Tanda-tangan digital kemudian di-*embed* bersama dengan pesan untuk disisipkan di dalam karya musik instrumental ciptaannya menggunakan steganografi.
5. Pihak inti 1 melakukan publikasi hasil karya musik instrumentalnya yang telah disisipkan dengan sebuah tanda-tangan digital yang telah di-*embed* dengan pesan.

Setelah itu, proses dilanjutkan dengan tahap validasi yang dilakukan oleh pihak inti 2, yaitu platform publikasi dengan melakukan proses verifikasi tanda-tangan digital.

1. Pihak inti 2, yaitu platform menerima *file* audio hasil steganografi karya musik instrumental dari pihak inti 1.
2. Pihak inti 2, melakukan *decode* atau ekstraksi tanda-tangan digital yang telah di-*embed* dalam pesan pada file audio.
3. Tanda-tangan digital yang telah diekstraksi kemudian didekripsi menggunakan kunci publik yang telah dibuat oleh pihak inti 1 sesuai kesepakatan dengan pihak inti 2 untuk menghasilkan *message-digest*. Misalkan disebut MD *A*.
4. Pesan yang berhasil diekstraksi bersamaan dengan tanda-tangan digital kemudian di-*hash* dengan metode *hash* yang sama yang digunakan oleh pihak inti 1 untuk menghasilkan sebuah *message-digest* lain. Misalkan disebut MD *B*.
5. Pihak inti 2 melakukan perbandingan antara MD *A*, dan MD *B*. Apabila didapati MD *A* = MD *B,* maka verifikasi berhasil dilakukan. Apabila MD *A* ≠ MD *B,* maka verifikasi tidak berhasil.

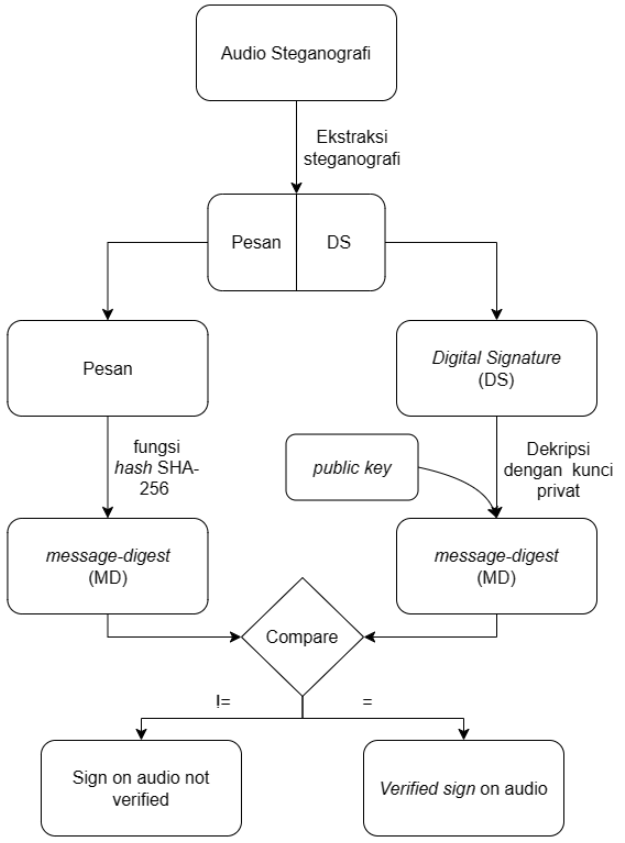
Oleh karena itu Secara umum, skema rancangan umum solusi dapat disusun seperti rancangan dibawah.





Gambar **4.** Rancangan skema penandatanganan sebelum publikasi

Verifikasi oleh platform



Gambar **5.** Rancangan skema verifikasi tanda tangan oleh platform

C. Implementasi Solusi Permasalahan

Solusi diimplementasi dengan beberapa tahapan, tahapan tersebut antara lain.

1. Pembuatan pasangan kunci RSA

Pertama-tama, perlu dibuat pasangan kunci yang terdiri dari kunci publik dan kunci privat. Pasangan kunci ini dibuat menggunakan sebuah *python library* yang bernama PyCryptodome. RSA — PyCryptodome 3.18.0 documentation dengan luaran berupa file dengan format .pem, yang berisi pasangan kunci

contoh pasangan kunci:

contoh kunci privat dalam file private_key.pem

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA150XQzn2NeuZBkLjbneF0Qfs1je
XfbUGTm2ZllKFWW1lID3C
5mbigPcwN9D6hzKzJ5ze0z+RFASy4gQ91BDYUjbmk7ce
hsg4vAnydhJGLep789mA…..
…..Ycp
-----END RSA PRIVATE KEY-----
```

contoh kunci publik dalam file public_key.pem

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKC
AQEA150XQzn2NeuZBkLjbneF
0Qfs1jeXfbUGTm2ZllKFWW1lID3C5mbigPcwN9D6hzK
zJ5ze0z+RFASy4gQ91BDY…..
……AQAB
-----END PUBLIC KEY-----
```

2. Proses *hashing*

Pada proses ini, pihak inti 1 membuat sebuah pesan input berupa pesan. Misalnya “lagu instrumen piano ini adalah ciptaan harith fakhiri”, atau pesan serupa lainnya. Kemudian pesan tersebut dilakukan proses *hashing* menggunakan fungsi *hash* yang dimiliki *library* PyCryptodome. SHA-256 — PyCryptodome 3.18.0 documentation. Contoh *message-digest* hasil *hash*.

```
67a52497ccaf1feba3ccaf5452d9a4f1ecb004ab02ba1deb025
6d5febba63c2f
```

3. Pembuatan tanda-tangan digital

Pembuatan tanda-tangan digital dilakukan dengan melakukan enkripsi pada hasil *message-digest*. Enkripsi ini dilakukan dengan menggunakan *library* PyCryptodome yang menyediakan fungsi untuk melakukan pembuatan tanda-tangan digital. PKCS#1 v1.5 (RSA) — PyCryptodome 3.18.0 documentation. Tanda-tangan digital kemudian di-*embed* pada pesan yang telah dibuat.

Contoh dari hasil tanda-tangan digital

```
'\xb5\xbc\x18\xeb\xc1\xc3#\xf9\xfd@\x98\xafM\x98\xdf&\
xaa\x95e\x1b\x81\x82Cy9\xb2vs\xfcg
```

a_\x0f\x8a\xccr2\xc2d<o3\xe3\xee?at1\x97\xc….

4. steganografi pada audio

Steganografi dilakukan dengan cara melakukan penyisipan pesan yang telah di-*embed* dengan tanda-tangan digital. Penyisipan dilakukan dengan melakukan modifikasi algoritma yang telah ditemukan pada sumber: https://github.com/sniperline047/Audio-Steganography/blob/master/stego.py

5. Proses verifikasi tanda-tangan digital

Verifikasi dilakukan dengan melakukan ekstraksi hasil pesan yang disisipkan dalam steganografi. Contoh hasil pesan yang disisipkan.

lagu instrumen piano ini adalah ciptaan harith fakhiri

<ds>'\xb5\xbc\x18\xeb\xc1\xc3#\xf9\xfd@\x98\xafM\x98\xdf&\xaa\x95e\x1b\x81\x82Cy9\xb2vs\xfcg
a_\x0f\x8a\xccr2\xc2d<o3\xe3\xee?at1\x97\xc…… <ds>

Pesan dan tanda-tangan digital kemudian dipisah. Pesan yang disisipkan di-*hash* kembali menggunakan fungsi *hash* SHA-256 seperti yang dilakukan pada saat pengiriman. Kemudian, tanda-tangan digital didekripsi dan diverifikasi dengan melakukan komparasi terhadap hasil digest. Verifikasi ini dilakukan menggunakan fungsi verify yang terdapat pada *library* PyCryptodome. PKCS#1 v1.5 (RSA) — PyCryptodome 3.18.0 documentation. Verifikasi dilakukan dengan melakukan *try and exception*.

```
try:
    pkcs1_15.new(key).verify(h, bytes(get_sign,'utf-8'))
    print("The signature is valid, message verified.")
    print("the message digest is : ", h.hexdigest())
    print()
    print("Pesan yang disisipkan : ")
    print(message)
    print("<ds>",get_sign,"<ds>")
except (ValueError, TypeError):
    print()
    print("The signature is not valid, message digest might be different or there are type errors")
```

IV. Hasil Pengujian dan Analisa

A. Pengujian

Pada tahap pengujian, dilakukan dua percobaan pada proses pengujian. Yaitu penggunaan kunci publik dan privat sesuai kesepakatan dan ketentuan, serta penggunaan kunci publik dan privat yang tidak sesuai kesepakatan dan ketentuan.

Pada pengujian pertama, dilakukan sebuah buat sebuah pasangan kunci publik menggunakan PyCryptodome yang sesuai. Contoh: Setelah itu, pasangan kunci di load sesuai dengan kesepakatan dan ketentuan. Contoh:

```
key = RSA.generate(2048)
private_key = key.export_key()
file_out = open("private_key.pem", "wb")
file_out.write(private_key)
file_out.close()

public_key = key.publickey().export_key()
file_out = open("public_key.pem", "wb")
file_out.write(public_key)
file_out.close()

## line of code....

priv_key =
RSA.import_key(open('private_key.pem').read())
pub_key =
RSA.import_key(open('public_key.pem').read())
```

Kemudian, masukkan dari audio adalah ‘sample.wav’ dan masukan pesan adalah “hak cipta milik harith”, Maka, dihasilkan luaran berupa:

```
bb\xfa\x18\x1c\x10h{\xfc\x04\x94v\xf0?I\xd9\x9b\x8f\x0f\xc5\xe8\\\x81e\x86\xa0G\xef\xcc!\xb9z\xc3

The signature is valid, message verified.
the message digest is :  057c6bffaf7636d9b96aa2671b94e2b788db20bc81e0845576aa39f406062870

Pesan yang disisipkan :
b'hak cipta milik harith'
<ds> y\x05\xd3\x8c\xae\xb2\x8e\x06\xff\x86\x10\x01\xc0\xcbN\xa9g\xf3x\x00G\x13k\xfe\x9d\x97\xb9\xd8E\x18;\xc3
\x9d\x87\xe2\x9a\xbb\xb4V@\xe1\xcd\xca\'\xa8v\xba\xfd+\xaa}^]\xd4\x13:bq=\tB\x84\xb6\xe5\xe9yx-5z\xf1\xe7\xb3
\xde\xb0A\x89\x93\xbfX\x8b\x7f\x8c{\xe7A\x19p\xfe\x11\xdf"\xe7\x978\x0f\xd7\xd2\xce\xc7C\xcbT\xf5\xfa`1\x1d \
x07\x8fh\xf5\x898\xb0~MFK\x83Neu\x1aS|\x8b\xf5\xb1\xb5Kw(\x8c\xba\xc1\xa8n\xd0\x07\xe0r\x05\xa1G<\xf5\x8f\xb4
\xf1\x825\x91.\xd1-\xe5~Rq}\xc9\xdf\xa1>2P`4]\x13\x06\xf3\xdfW\x0cv\xe1\xff\x86\xcd\xa0\xe5\x1fF\x0f\xeeK\x0f
n/R(\x9c\xf4\x7f \xf7\x9e5\x1et"\x93\xf7\x0e\xf4\x85o\xe4\x8f\xc2h\xb1\xe1!@vL\xdc2\xe7\x82\xbb\xfa\x18\x1c\x
10h{\xfc\x04\x94v\xf0?I\xd9\x9b\x8f\x0f\xc5\xe8\\\x81e\x86\xa0G\xef\xcc!\xb9z\xc3 <ds>
```

Gambar **6.** Pengujian satu

Kemudian pada pengujian kedua, misal kunci publik yang digunakan tidak sesuai kesepakatan. Contohnya

```
key = RSA.generate(2048)
private_key = key.export_key()
file_out = open("private_key.pem", "wb")
file_out.write(private_key)
file_out.close()

key1 = RSA.generate(2048)
private_key = key.export_key()
file_out = open("private_key_other.pem", "wb")
file_out.write(private_key)
file_out.close()
```

```
public_key = key1.publickey().export_key()
file_out = open("public_key.pem", "wb")
file_out.write(public_key)
file_out.close()

## line of code....

priv_key =
RSA.import_key(open('private_key.pem').read())
pub_key =
RSA.import_key(open('public_key.pem').read())
```

Kemudian, masukkan dari audio tetaplah sama, yaitu ‘sample.wav’ dan masukan pesan adalah “hak cipta milik harith”, Maka, dihasilkan luaran berupa:

```
decoded sign: \x1e3nQ\xdaU\xe9\xc5\xe2\x03\xe6\xd1\x04.6\xf4\x1d\xe5?\xdf[?\x8a\xaap\xa5
c\xd1r\x94\xdf\xc4L\x868\x87\x12f\xff\xb5\xa9\xe0N\xd3\x9aO\xb3M\x13C\x0c\xe1\x10\xce\x0
6\xaa\x04D\xcd\x14\xfc\xe0\xae\xaa[\xb6\xf1\xfe?\xbb<\x12i\xcb\xd8p\xac 9\x80\xdd.PU\xca
\xedj[\n0\x8c\xc4\xbf\x87"\xb4\xbbt\r\xc3\xe6\xc3 \x7f\x13\xbe\x08\xe7\xc0k\x97\xbe\x96)
c8\xb6\xf4\x12.,\x90S\xf8\n(\xdb\xc1\xdcy\xbe\xb3\xb8\n:\xaeW\xdc\x99\x029\xa5\xe2\x06\x
\xe5R\x8c\x94\xab\xfbcT\xb7\xc1\xa0\xdb\x88\xb7\xc5Md\x97\x81a\xe7\xef\x18\xc2V\xcf\x8d\
f\xcaf\x13=\xeb:\xe3\xfd\x90\x8f\xee\xd0b\xc0Ymj7\xf0\xa2_c\x91*X)P\x104\xf4\xc9\xae\xe0
7

The signature is not valid, message digest might be different or there are type errors
PS C:\Users\Lenovo\Documents\semester 8\kripto\tugas makalah\Audio-Steganography>
```

Gambar **7.** Pengujian dua

Kegagalan verifikasi ini terjadi akibat kesalahan pada saat melakukan *load* sehingga kunci publik yang digunakan bukanlah dibangkitkan dari kunci privat yang digunakan.

B. Analisa

Pada pengujian pertama, verifikasi dengan pembuatan pasangan kunci privat dan kunci publik yang sesuai menghasilkan luaran yang valid, yaitu berhasil terverifikasinya hak milik pihak inti 1 pada audio yang sudah disisipkan pesan dengan steganografi.

Kemudian pada pengujian kedua, verifikasi dengan pembuatan pasangan kunci privat dan kunci publik yang tidak sesuai, dimana kunci publik dibangkitkan dari kunci privat lain, menghasilkan luaran berupa gagalnya verifikasi.

Oleh karena itu, dapat disimpulkan bahwa dalam penandatanganan digital dengan kombinasi kriptografi kunci publik dan fungsi *hash* harus, pasangan kunci yang dibuat sangat lah penting. Apabila terjadi kesalahan dalam pembuatan pasangan kunci, maka dapat menyebabkan terjadinya kesalahan pada verifikasi tanda-tangan digital.

V. Kesimpulan dan Saran

Dari penelitian yang telah dilakukan pada makalah ini, dapat dibuktikan bahwa implementasi tanda-tangan digital dengan kombinasi kriptografi kunci publik, fungsi *hash*, serta audio steganografi pada karya musik instrumental dapat membantu musisi yang belum memiliki label untuk menjaga hak cipta nya walau karya nya sudah terpublikasi.

Kedepannya, penulis dapat meningkatkan solusi penelitian dengan membuat skenario serta uji coba apabila terdapat pihak eksternal yang berusaha merebut hak milik karya musik instrumental yang bukan miliknya.

Ucapan Terimakasih

Penulis mengucapkan rasa syukur kepada Tuhan Yang Maha Esa karena dengan izin dan berkat-Nya penulis dapat menjalani mata kuliah IF4020 Kriptografi ini dari awal sampai pembuatan makalah ini. Terima kasih juga diucapkan kepada Bapak Rinaldi Munir sebagai dosen pengampu mata kuliah ini yang telah mengajarkan kepada penulis materi yang diperlukan dalam pembuatan makalah ini.

References

[1] Munir, R. "Tanda-tangan digital (digital signature)". 2022. Retrieved Mei 21, 2022, from, https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2022-2023/28-Tanda-tangan-digital-2023.pdf

[2] A. Anand, “Breaking Down : SHA-256 Algorithm,” Medium, 13 Januari 2020.https://infosecwriteups.com/breaking-down-sha-256-algorithm-2ce61d86f7a3

[3] Munir, R. “Fungsi Hash”. 2022. Retrieved Mei 22, 2022, from, https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2022-2023/25-Fungsi-hash-2023.pdf

[4] Munir, R. “Algoritma RSA”. 2022. Retrieved Mei 22, 2022, from, https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2022-2023/19-Algoritma-RSA-2023.pdf

[5] Munir, R. “Steganografi”,. 2022. Retrieved Mei 22, 2022, from, https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2022-2023/08-Steganografi-Bagian1-2023.pdf

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, 22 Mei 2023

Harith Fakhiri Setiawan
13519161