Implementasi *Digital Signature* pada *AI Generated Cover Song*

Fathan Ananta Nur (18219008)
Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: 18219008@std.stei.itb.ac.id

***Abstract*— Musik telah menjadi bagian penting dalam kehidupan manusia dan terus berkembang seiring dengan kemajuan teknologi. Salah satu teknologi yang terus berkembang dalam dunia musik adalah penggunaan *artificial intelligence* (AI) untuk menciptakan lagu-lagu baru atau membuat cover lagu-lagu yang sudah ada. Namun, hal ini membawa tantangan baru dalam hal perlindungan hak cipta dan keaslian musik yang dihasilkan oleh AI. Implementasi *digital signature* pada *AI Generated Cover Song* menjadi fokus utama makalah ini. Makalah ini akan membahas pembuatan *cover song* berbasis AI, penggunaan *digital signature* untuk memastikan validitas dan integritas musik buatan AI, serta pengujian keaslian musik yang dihasilkan oleh AI menggunakan *digital signature*. Makalah ini bertujuan untuk mengeksplorasi cara mengimplementasikan *digital signature* sebagai langkah penting dalam memastikan keaslian dan keabsahan musik yang dihasilkan oleh AI.**

***Keywords*—lagu; *hash*; enkripsi; hak cipta**

I. PENDAHULUAN

Perkembangan teknologi *artificial intelligence* (AI) telah mengubah banyak aspek kehidupan, termasuk di industri musik. Salah satu aplikasi menarik dari AI adalah kemampuannya dalam menghasilkan *cover song* secara otomatis. Namun, dengan adanya karya musik yang dihasilkan oleh AI, muncul pula pertanyaan tentang hak cipta dan keabsahan legalitas karya tersebut. Dalam konteks *cover song*, hak cipta sangat penting untuk melindungi karya asli dan memberikan penghargaan kepada penciptanya. Meskipun AI dapat menghasilkan *cover song* yang sangat mirip dengan aslinya, tetap saja ada kebutuhan untuk memastikan bahwa legalitas hak cipta tetap terjaga dan identifikasi pencipta asli tetap terlacak.

Untuk mengatasi tantangan ini, implementasi tanda tangan digital menjadi solusi yang relevan. Tanda tangan digital adalah teknologi yang memungkinkan identifikasi dan verifikasi keaslian suatu dokumen atau *file* secara elektronik. Dengan menerapkan tanda tangan digital pada musik yang dihasilkan oleh AI, kita dapat memberikan legalitas yang diperlukan dan mengidentifikasi pencipta asli dari karya tersebut. Makalah ini akan membahas tentang implementasi tanda tangan digital pada *AI Generated Cover Song*, dengan fokus pada pentingnya menjaga hak cipta, memastikan legalitas karya, dan mengidentifikasi pencipta asli. Makalah ini akan menjelaskan konsep tanda tangan digital dan bagaimana penerapannya pada musik yang dihasilkan oleh AI. Selain itu, makalah ini juga akan membahas pengujian keaslian musik yang dihasilkan oleh AI menggunakan tanda tangan digital.

Melalui makalah ini, diharapkan pemahaman tentang pentingnya implementasi tanda tangan digital pada musik yang dihasilkan oleh AI dapat meningkat. Dengan adanya tanda tangan digital, kita dapat menjaga hak cipta, memastikan legalitas karya, dan memberikan pengakuan yang pantas kepada pencipta asli dalam dunia *AI Generated Cover Song*.

II. METODE

Dalam mengembangkan implementasi *digital signature* pada *AI Generated Cover Song*, digunakan metode sebagai berikut.

A. Studi Literatur

Pada langkah awal, akan dilakukan studi literatur berkaitan dengan topik makalah, yaitu algoritma *digital signature* dan AI untuk membuat *cover song* berdasar data yang sudah tersedia. Dari bagian ini, akan dihasilkan mekanisme, struktur, dan informasi dari topik terkait sebagai dasar dalam pengembangan lebih lanjut.

B. Implementasi

Berdasarkan rancangan yang sudah dirumuskan, akan diimplementasi sebuah program untuk membubuhkan tandatangan digital sesuai dengan *file* audio *cover song* yang dibuat menggunakan AI. Selain itu, juga dijelaskan lingkungan implementasi, seperti bahasa pemrogram, *library*, dsb. yang digunakan untuk mengembangkan program tersebut. Implementasi hanya difokuskan kepada pemasangan *digital signature* di *file* audio saja tanpa menjelaskan implementasi pembuatan *cover song* menggunakan AI.

C. *Testing*

Metode terakhir yaitu pengujian dari program yang sudah dikembangkan. Pengujian dilakukan berdasarkan beberapa *testcase*, serta akan dievaluasi hasilnya.

III. STUDI LITERATUR

Bab ini akan membahas studi literatur yang relevan terkait implementasi *digital signature* pada *AI Generated Cover Song*. Melalui pemahaman mendalam tentang konsep dan prinsip dasar, akan diperkenalkan dengan landasan teoretis yang melandasi penggunaan *digital signature* dalam konteks menciptakan dan mengidentifikasi keaslian *cover song* yang dihasilkan oleh *artificial intelligence*.

A. *Cover Song*

Musik populer telah menjadi fenomena budaya yang sangat signifikan dalam kurun waktu abad ke-20 dan ke-21 [1]. Perkembangan media penyiaran, sistem perekaman yang baru, dan munculnya kelas menengah telah memfasilitasi konsumsi massal musik rekaman, menjadikannya sebagai salah satu industri hiburan yang paling sukses saat ini [2]. Salah satu elemen yang mengungkapkan sifat warisan dari musik populer adalah adanya *cover*, adaptasi, atau *remake* dari lagu-lagu [3]. Sebuah *cover* adalah rekaman baru dari sebuah lagu yang awalnya ditulis atau dibawakan oleh musisi lain. Menurut Serrà [4], versi *cover* adalah interpretasi alternatif dari sebuah lagu yang sebelumnya direkam. Mengingat bahwa sebuah *cover* dapat berbeda dari lagu aslinya dalam hal timbre, tempo, struktur, kunci, susunan, atau bahasa vokal. Menurut Tsai [5], versi *cover* sebuah lagu merujuk pada interpretasi baru dari sebuah lagu yang awalnya direkam dan populer oleh seorang artis lain.

B. Implementasi Hak Cipta pada Karya *Cover Song*

Hak cipta dalam sebuah komposisi musik awalnya dimiliki oleh penciptanya, yaitu komposer dan penulis lirik. Akan tetapi, biasanya para penulis lagu akan mentransfer hak cipta mereka kepada penerbit musik yang akan membantu mempromosikan lagu, mengelola pembayaran royalti, dan melindungi hak cipta [6]. Setelah sebuah karya musik telah diterbitkan, siapapun dapat merekam versi *cover* dari lagu tersebut dengan mendapatkan lisensi mekanik. Sebuah lagu dianggap diterbitkan ketika salinan atau rekaman lagu tersebut didistribusikan kepada masyarakat untuk dijual atau disewakan. Penampilan langsung tidak dianggap sebagai penerbitan. Di Indonesia, pemberian lisensi ini diatur dalam Undang-Undang Republik Indonesia No. 19 Tahun 2002 Tentang Hak Cipta [7].

Konsekuensi dari mempos sebuah *cover song* tanpa lisensi musik bergantung pada pemegang hak cipta. Beberapa pemegang hak cipta tidak masalah dengan adanya *cover song* di YouTube. Hal tersebut dapat meningkatkan eksposur lagu dan memperkenalkan audiens baru terhadap musik pencipta atau penampil asli. Jika lagu-lagu diposting oleh para penggemar, sebuah band tidak akan cenderung mengambil risiko merugikan mereka dengan menghapus video mereka. Namun, pemegang hak cipta lainnya mungkin keberatan dengan penggunaan tanpa lisensi atas karyanya.

C. *AI Cover Song*

AI *cover song* adalah teknologi yang menggunakan algoritma pembelajaran mesin untuk membuat versi *cover* dari lagu apa pun. Teknologi ini memungkinkan untuk menggantikan suara penyanyi asli dengan suara penyanyi lain, bahkan suara diri sendiri. Teknologi *AI cover song* menggunakan teknologi *neural network* untuk menganalisis struktur lagu asli dan menghasilkan versi baru yang terdengar mirip dengan yang asli [8].

D. Algoritma RSA

Algoritma RSA merupakan salah satu algoritma kriptografi yang paling populer dalam keamanan jaringan. Algoritma ini ditemukan oleh Ron Rivest, Adi Shamir, dan Leonard Adleman pada tahun 1977 di MIT. RSA banyak digunakan dalam keamanan jaringan. Pada algoritma ini, digunakan dua bilangan prima besar. Terdapat dua ide utama dalam RSA, yaitu masalah faktorisasi bilangan bulat dan masalah RSA, yaitu mencari akar N. N merupakan hasil perkalian dua bilangan prima. RSA menggunakan dua kunci, yaitu kunci publik dan kunci privat. Dalam tandatangan digital, kunci publik digunakan hanya untuk dekripsi, sedangkan kunci privat digunakan untuk enkripsi. Menurut teori bilangan, mudah untuk menghitung hasil perkalian dua bilangan besar, tetapi faktorisasi sulit dilakukan. Keamanan RSA bergantung pada faktorisasi bilangan-bilangan besar tersebut. Ukuran kunci dalam algoritma RSA adalah 2048 hingga 4096, yang sulit untuk difaktorisasi [9].

Terdapat 3 proses utama dalam algoritma RSA pada tandatangan digital, yaitu: *key generation*, *encryption*, dan *decryption*. Di bawah ini merupakan algoritma ketiga proses tersebut secara garis besar.

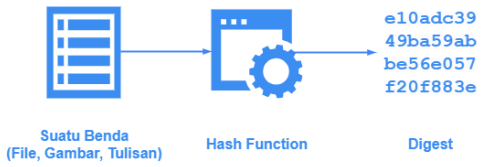
1. *Key generation*
   - Definisikan dua bilangan prima acak besar p dan q, dengan FPB p dan q = 1.
   - Hitung n = p * q
   - Hitung *totient euler* n dengan (p-1)*(q-1)
   - Pilih integer kunci publik e, untuk 1 < e < totient n, FPB n dan *totient* n adalah 1.
   - Hitung integer kunci privat d, e * d(mod(totient N)) = 1
   - Dihasilkan kunci publik adalah (e, n) dan kunci privat adalah (d, n)
2. Enkripsi
   - Menggunakan kunci privat (d, n)
   - Enkripsi pesan dalam *cipher value* c = m^d mod n.
3. Dekripsi
   - Menggunakan kunci publik (e, n)

- Enkripsi pesan dalam *cipher value* m = c^e mod n.

E. *Keccak Hash Function* (SHA-3)

Dalam kriptografi, fungsi *hash* adalah sebuah algoritma matematis yang digunakan untuk mengubah data dengan ukuran sembarang (*message*) menjadi sebuah *array* dengan ukuran tetap (sering disebut sebagai *hash value* atau *message digest*) [10]. Fungsi *hash* bertujuan untuk mengkompresi data sehingga menghasilkan representasi yang unik dan konsisten, yang dapat digunakan untuk memverifikasi integritas data dan mengidentifikasi perubahan atau manipulasi pada data tersebut.

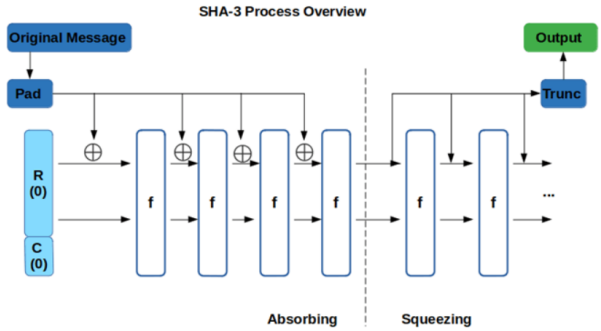
Fungsi *hash* ini memiliki sifat yang tidak dapat dibalik, artinya nilai *hash* tidak dapat dikembalikan ke bentuk aslinya atau di-*reverse* (sering disebut sebagai fungsi satu arah atau *one-way function*). Idealnya, setiap pesan akan menghasilkan *message digest* yang berbeda, sehingga berbagai jenis fungsi *hash* banyak digunakan dalam implementasi keamanan kriptografi modern. Fungsi *hash* ini bertujuan untuk memberikan representasi yang unik dan konsisten dari sebuah pesan, yang dapat digunakan untuk memverifikasi integritas data dan mendeteksi perubahan atau manipulasi pada pesan tersebut.



Gambar 1. Sistem Kerja Fungsi *Hash*

SHA-3 atau Keccak adalah jenis fungsi *hash* yang menjadi pengembangan dari fungsi SHA-1 dan SHA-2. Algoritma ini diciptakan oleh tim yang terdiri dari Guido Bertoni, Joan Daemen, Michaël Peeters, dan Gilles Van Assche, dan resmi dirilis oleh NIST (National Institute of Standards and Technology) pada tanggal 5 Agustus 2015.

SHA-3 menggunakan konstruksi sponge yang memungkinkan penyesuaian panjang nilai *hash* (*message digest*). Proses *hash* dimulai dengan proses *padding* pada pesan untuk menghasilkan ukuran yang sesuai dengan panjang *output* yang diinginkan. Selanjutnya, pesan yang telah di*padding* dibagi menjadi blok-blok dengan ukuran r-bit [11]. Informasi lebih lanjut mengenai panjang *output* standar dan nilai r dapat ditemukan di dokumentasi algoritma SHA-3.



Gambar 2. Algoritma SHA-3 [13]

Perlu diperhatikan bahwa total panjang nilai r+c harus mencapai 1600 bit. Pada tahap awal, *state* blok dengan ukuran r+c bit diatur menjadi 0, kemudian dilanjutkan ke dua tahap utama yaitu tahap penyerapan (*absorbing*) dan tahap pemerasan (*squeezing*) [12]. Tahap penyerapan bekerja dengan mengambil setiap blok pesan masukan dan memprosesnya melalui fungsi permutasi. Pada tahap pemerasan, *message digest* dengan panjang sesuai dengan *output* length awal dihasilkan.

IV. IMPLEMENTASI

A. Lingkungan yang Digunakan

Implementasi pelatihan data corpus audio dan rekonstruksi ulang lagi hasil AI menggunakan PyDub. PyDub adalah sebuah *library* Python yang digunakan untuk memanipulasi *file* audio. PyDub dapat melakukan tugas-tugas umum pada pemrosesan audio, seperti mengubah volume, menambahkan efek suara, memotong bagian-bagian tertentu dari *file* audio, dan sebagainya. Pembuatan model menggunakan algoritma SVC (*Support Vector Classification*) yang membedakan antara kelas audio yang berbeda berdasarkan fitur-fitur yang diekstraksi dari data audio tersebut. Akan tetapi, pada makalah ini, pembahasan mengenai pembentukan audio AI tidak dibahas lebih lanjut. Makalah ini lebih berfokus kepada penerapan dan manfaat *digital signature*/kriptografi dalam audio AI tersebut.

Di lain sisi, implementasi pembangkitan dan verifikasi tanda tangan digital dengan algoritma RSA dan SHA3-Keccak dalam program menggunakan bahasa Python. Hal ini dikarenakan adanya beragam *library* yang tersedia dalam bahasa pemrograman Python yang komprehensif, mudah digunakan, dan efisien dalam pengembangan program tersebut. Selain itu, algoritma SHA3 yang diterapkan dalam program ini memiliki panjang *output* sebesar 256 bit (SHA3-256) sehingga faktor keamanan dapat ditingkatkan.

Berikut ini *library* yang digunakan dalam pengembangan program pembangkitan dan verifikasi tandatangan digital.

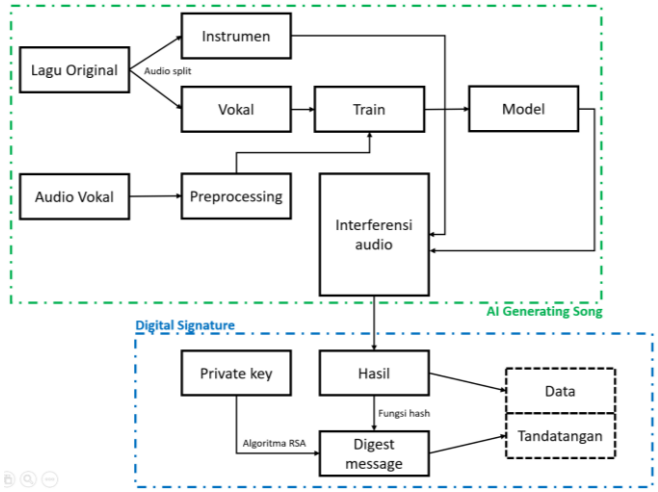
1. hashlib
   Library ini digunakan untuk melakukan *hash* menggunakan algoritma SHA3 256 bit (SHA3_256).
2. io
   Library ini merupakan modul standar yang menyediakan berbagai fungsi dan kelas untuk

mengelola operasi *input*/*output* (I/O), termasuk membaca dan menulis data dalam berbagai format.

3. random
   Library ini merupakan modul bawaan yang menyediakan berbagai fungsi untuk menghasilkan bilangan acak.

B. Pembangkitan Tandatangan Digital *AI Generated Cover Song*

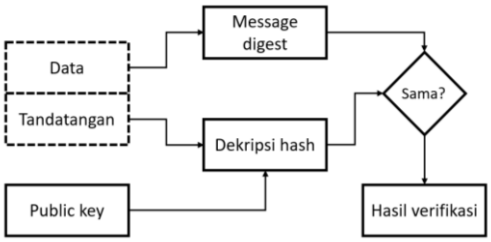
Gambar 3 di bawah merupakan langkah pembuatan *AI generated cover song* dan pembangkitan *digital signature* pada *file* audio tersebut.



Gambar 3. Diagram Pembuatan *AI Generated Cover Song* dan Tandatangan Digital

C. Verifikasi Tandatangan

Gambar 4 di bawah merupakan langkah untuk melakukan verifikasi tandatangan digital pada *AI Generated Cover Song*.



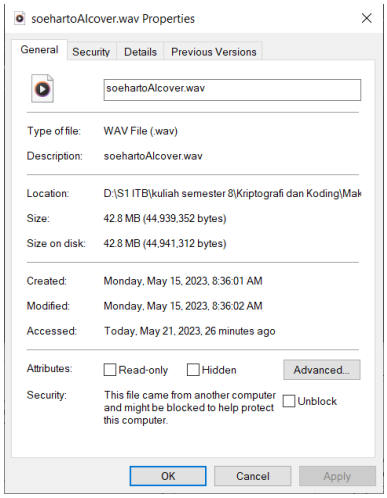
Gambar 4. Diagram Verifikasi Tandatangan Digital

V. *TESTING* DAN PEMBAHASAN

Pengujian terhadap program difokuskan kepada *digital signature* yang meliputi pembuatan sepasang kunci, pembubuhan tandatangan digital, dan verifikasi tandatangan digital. Pengujian dilakukan dengan menggunakan beberapa skenario *testcase*. *Testcase* meliputi fungsi generate key, *digital signature*, dan verifikasi.

Untuk melakukan menjalankan *testcase*, dibuatkan *file AI Generated Cover Song* menggunakan program yang sudah dirancang untuk diberikan tandatangan digital. *File* merupakan audio dari suara Presiden Soeharto yang menyanyikan/mengcover lagu *Renai Circulation* yang dibawakan oleh Kana Hanazawa. Berikut merupakan link YouTube dari audio tersebut: https://youtu.be/aGbqAM64duI.

Gambar 5 berikut merupakan detail dari *file* audio yang akan dilakukan *testing*.



Gambar 5. Audio untuk *Testing* Sistem

Berikut merupakan Tabel 1 yang menjelaskan beberapa skenario *testcase* yang akan dilakukan.

Tabel 1. *Testcase Item*

No	Deskripsi	Ekspektasi	Hasil
TC-1	Program membuat *private key* sesuai dengan spesifikasi algoritma RSA	*File* dengan ekstensi .pri berhasil disimpan.	Sukses
TC-2	Program membuat *public key* sesuai dengan spesifikasi algoritma RSA	*File* dengan ekstensi .pub berhasil disimpan.	Sukses
TC-3	Program menghasilkan *file* tandatangan digital sesuai dengan *file* yang di*input*kan untuk diberikan tandatangan digital	*File* dengan ekstensi .txt berhasil dihasilkan di *folder signature*. *File* text tersebut juga memiliki format yang sesuai.	Sukses
TC-4	Program melakukan verifikasi pada *file* dan *signature* yang sesuai.	*File* terverifikasi.	Sukses
TC-5	Program melakukan verifikasi pada *file* dan *signature* yang tidak sesuai.	*File* tidak terverifikasi.	Sukses
TC-6	Program melakukan verifikasi pada *file* dan *signature* yang	*File* tidak terverifikasi.	Sukses

	sesuai, tetapi menggunakan kunci publik yang salah.		

Berikut merupakan penjelasan dan detail untuk setiap *testcase*.

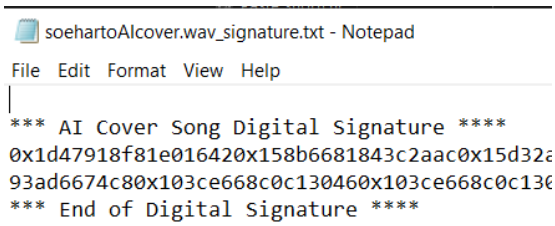
1. TC-1 dan TC-2

   Pengujian pada kedua *testcase* ini dilakukan untuk melihat apakah program dapat menghasilkan pasangan kunci publik dan kunci privat sesuai parameter yang dimasukkan. Kunci publik dan kunci primer berisikan nilai *tuple* yang harus sesuai dengan spesifikasi algoritma RSA. Kunci privat berisikan *tuple* (d, n) dan kunci publik berisikan *tuple* (e, n). Gambar 6 berikut merupakan isi dari kunci privat dan kunci publik yang dihasilkan di *testcase* ini.

Gambar 6. Hasil TC-1 dan TC-2

2. TC-3

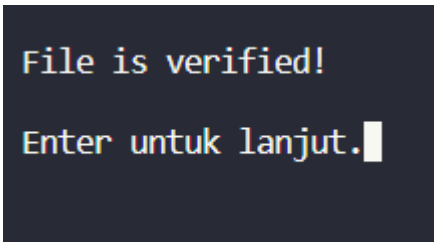
   Pengujian ini dimaksudkan untuk melihat apakah program dapat menghasilkan *file* baru berupa *file text document* yang menyimpan *hash value* dari tandatangan sebuah *file* yang di*input*kan ke program. Struktur teks di dalam *file text document* tersebut juga harus sesuai dengan format yang ditentukan program, yakni diawali dengan string tertentu, kemudian diisikan dengan *hash value* tandatangan digital, dan kemudian ditutup dengan string tertentu. Gambar 7 berikut merupakan isi dari *file text document* hasil tandatangan digital dari *file* yang dimasukkan ke program.



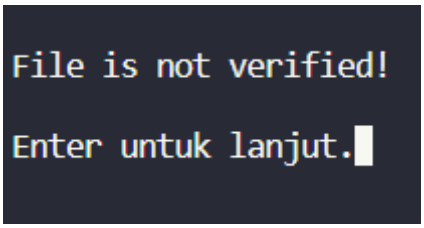
Gambar 7. Hasil TC-3

3. TC-4, TC-5, dan TC-6

   Pengujian ini dilakukan apakah program dapat memberikan verifikasi pada sebuah *file* dengan tandatangan digital tertentu. Hasil verifikasi ditentukan dengan sama tidaknya *hash value* sebuah *file* dengan hasil deksripsi *hash value* tandatangan digital. Dekripsi juga harus menggunakan kunci publik yang berpasangan dengan kunci privat saat penandatangan digital dilakukan di *file* terkait. Berikut Gambar 8 dan 9 yang menunjukkan hasil verifikasi sesuai dengan *testcase* yang dijalankan.



Gambar 8. Hasil TC-4



Gambar 9. Hasil TC-5 dan TC-6

VI. KESIMPULAN DAN SARAN

Dari hasil implementasi dan *testing* yang dilakukan, dapat disimpulkan beberapa poin berikut.

1. Penggunaan algoritma RSA dan SHA-3 Keccak dapat digunakan sebagai alternatif untuk pemberian tandatangan digital pada *AI generated cover song*, sehingga dapat memberikan identifikasi pada sebuah AI *cover song*. Hal ini juga mempermudah pengimplementasian hak cipta pada sebuah lagu yang diciptakan oleh mesin, dengan catatan yang akan dituliskan di bagian saran.
2. Keamanan yang diberikan oleh SHA-3 berbeda-beda karena penerapan metode konstruksi sponge yang membutuhkan pendekatan kriptanalisis yang baru. Selain itu, panjang *hash value* dapat divariasikan sehingga dapat disesuaikan dengan kinerja sistem yang sesuai.

Dari beberapa kesimpulan di atas, disimpulkan juga beberapa saran untuk pengembangan kedepannya maupun pengembangan lingkungan *AI generated cover song* baik diimplementasi maupun aspek lainnya.

1. Setiap program AI yang menghasilkan produk yang sarat akan legalitas, sebaiknya diberikan fitur *digital signature* sehingga mempermudah proses *declare* maupun langkah prosedural lainnya ketika dihadapkan ke ranah legal.
2. Pemberian *digital signature* yang dibahas di makalah ini, baru sebatas memberikan *signature* yang terpisah dari *file* utama. Diharapkan ada eksperimen selanjutnya yang dapat mengimplementasikan tandatangan ke *file binary* yang juga tercantum di dalamnya, sehingga lebih memperkuat aspek keamanan dan non-repudiasi.

LINK SOURCE CODE GITHUB

https://github.com/thefathan/AICoverSignature

DAFTAR PUSTAKA

[1] Shuker R, "Understanding popular music culture," 5th edition, London: Routledge, 2016

[2] Burnett R, "The global jukebox: The international music industry," Psychology Press, 1996.

[3] Wall T, "Studying popular music culture," London: Sage, 2013.

[4] Serrà J, Gómez E, and Herrera P, "Audio Cover Song Identification and Similarity: Background, Approaches, Evaluation, and Beyond," Advances in Music Information Retrieval, Springer Berlin Heidelberg, 2010, pp. 307-322.

[5] Tsai W, Yu H, and Wang H, Using the Similarity of Main Melodies to Identify Cover Versions of Popular Songs for Music Document Retrieval, J. Inform. Sci. Eng, 2008, 24:1669-1687.

[6] H. Black, "Posting cover songs on YouTube? what you need to know," LegalZoom, https://www.legalzoom.com/articles/posting-cover-songs-on-youtube-what-you-need-to-know (accessed May 21, 2023).

[7] Pemerintah Pusat, "Hak Cipta," UU No. 19 tahun 2002 Tentang Hak Cipta [JDIH bpk ri], https://peraturan.bpk.go.id/Home/Details/44465/uu-no-19-tahun-2002 (accessed May 21, 2023).

[8] Daniel, "Ai song cover generator: How to cover ariana grande song with Drake Ai Voice," TopMediai, https://www.topmediai.com/text-speaker/ai-song-cover/ (accessed May 21, 2023).

[9] A. Saini and D. Vandana, "A STUDY ON MODIFIED RSA ALGORITHM IN NETWORK SECURITY," vol. 4, pp. 1461–1465, 04 2022.

[10] R. Munir, "Fungsi Hash," [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi-danKoding/2021-2022/17%20-%20Fungsi-hash-2021.pdf. (accessed May 21, 2023).

[11] R. Munir, "SHA-3 (Keccak)," [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi-danKoding/2021-2022/20%20-%20SHA-3-2020 (accessed May 21, 2023).

[12] A. Anand, "Breaking down : SHA-3 algorithm," Medium, 13-Jan-2020. [Online]. Available: https://infosecwriteups.com/breaking-down-sha-3-algorithm-70fe25e125b6. (accessed May 21, 2023).

[13] Admin, "Hash algorithm comparison: MD5, SHA-1, SHA-2 & sha-3," Code Signing Store, https://codesigningstore.com/hash-algorithm-comparison (accessed May 21, 2023).

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

Ttd

Fathan Ananta Nur

18219008