Penerapan Tanda Tangan Digital dan Steganografi untuk Verifikasi Status Legal Komik Digital

Fithratulhay Pribadi and 13517140[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13517140@std.stei.itb.ac.id

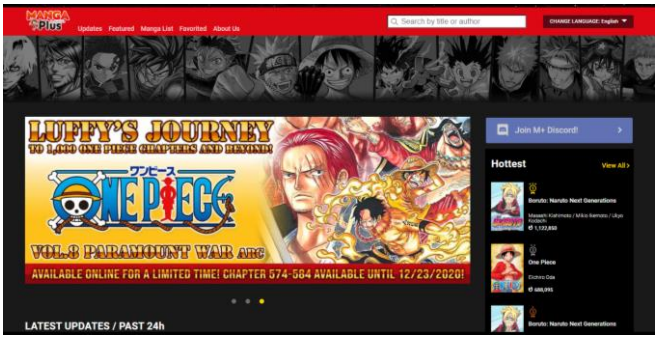
***Abstract*— Sulit bagi pembaca komik digital untuk mengetahui apakah situs yang ia gunakan adalah situs yang telah mempunyai izin untuk menyebarkan komik secara legal. Untuk mempermudah pembaca dalam mengetahui apakah situs yang ia gunakan legal atau tidak, diperlukan sebuah sistem yang dapat memberikan verifikasi secara otomatis mengenai status legal sebuah komik yang ada di suatu situs. Sistem yang disarankan oleh penulis pada makalah ini adalah sistem yang memanfaatkan tanda tangan digital.**

***Keywords*— Komik, legal, steganografi, tanda tangan digital.**

I. PENDAHULUAN

Komik adalah bentuk kartun yang mengungkapkan karakter dan memerankan suatu cerita dalam urutan yang erat dihubungkan dengan gambar dan dirancang untuk memberikan hiburan kepada pembacanya. Dahulu, komik disebarkan oleh penerbit dalam bentuk buku secara fisik. Namun, seiring berkembangnya teknologi, ada banyak pihak yang mulai memindai komik yang ia beli dan menyebarkannya melalui situsnya sendiri untuk meraup untung secara ilegal.

Untuk melawan penyebaran ilegal tersebut, penerbit mulai melakukan cara untuk menyebarkan komik secara legal melalui situs yang dibuat oleh penerbit tersebut atau situs lain yang telah memiliki izin secara legal untuk menyebarkannya. Sebagai contoh, penerbit Kodansha USA, sebuah penerbit komik jepang di Amerika, menyebarkan komiknya melalui aplikasi INKR Comics, Mangamo, dan Izneo. Penerbit komik lain di Amerika seperi VIZ Media juga menyebarkan komiknya secara digital melalui aplikasi VIZ Manga dan Shonen Jump Manga & Comics. Penerbit komik dari Jepang seperti Shueisha pun turut menyebarkan komiknya melalui situs dan aplikasi Manga Plus. Contoh tampilan situs komik Manga Plus dapat dilihat pada Gambar 1.1.



Gambar 1.1. Manga Plus

Meskipun telah ada situs ataupun aplikasi yang telah menyediakan komik secara legal, masih ada banyak situs ilegal yang ada di internet. Karena itu, terkadang sulit bagi pembaca untuk mengetahui apakah situs yang ia gunakan adalah situs yang telah mempunyai izin untuk menyebarkan komik secara legal. Untuk memastikan hal tersebut, biasanya pembaca perlu mencari-cari sendiri informasi tentang apakah situs yang ia gunakan untuk membaca merupakan situs yang legal atau tidak.

Untuk mempermudah pembaca dalam mengetahui apakah situs yang ia gunakan legal atau tidak, diperlukan sebuah sistem yang dapat memberikan verifikasi secara otomatis mengenai status legal sebuah komik yang ada di suatu situs. Sistem yang disarankan oleh penulis pada makalah ini adalah sistem yang memanfaatkan tanda tangan digital.

II. LANDASAN TEORI

A. Komik Digital

Pada saat ini, komik digital yang ada di internet terbagi menjadi dua macam, yaitu komik ilegal dan komik legal. Komik legal merupakan komik yang disebarkan di internet tanpa izin penerbit asal. Pada umumnya, komik ini merupakan hasil pemindaian oleh orang yang telah membeli komik fisik.

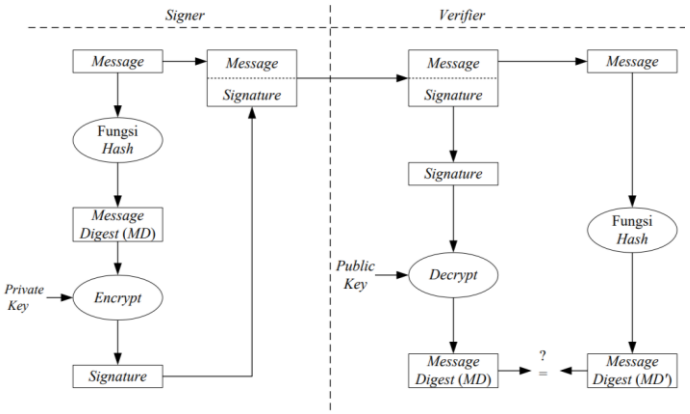
Bersebrangan dengan hal tersebut, komik legal adalah komik yang disebarkan di internet dengan izin dari penerbit. Terkadang, pihak penerbit mengembangkan sendiri aplikasi untuk menyebarkan komik secara digital. Selain itu, ada juga pihak yang membayar lisensi kepada penerbit sehingga mendapatkan izin untuk menyebarkannya secara digital.

B. Tanda Tangan Digital

Tanda tangan adalah tanda sebagai lambang nama yang dituliskan dengan tangan oleh orang itu sendiri sebagai penanda pribadi (telah menerima dan sebagainya). Sejak dulu, tanda tangan sudah digunakan untuk otentikasi dokumen cetak. Tanda tangan memiliki karakteristik sebagai berikut :

1. Tanda tangan adalah bukti yang otentik
2. Tanda tangan tidak dapat dilupakan
3. Tanda tangan tidak dapat dipindah untuk digunakan ulang
4. Dokumen yang telah ditandatangani tidak dapat diubah
5. Tanda tangan tidak dapat disangkal

Tanda tangan digital adalah tanda tangan yang digunakan untuk data digital. Berbeda dengan tanda tangan biasa, tanda tangan digital bukan merupakan tulisan tanda tangan yang di-digitasi dengan cara dipindai atau difoto, melainkan nilai kriptografis yang bergantung pada isi pesan dan kunci. Jika tanda tangan pada dokumen cetak selalu sama untuk setiap dokumen, tanda tangan digital akan berbeda-beda tergantung isi dokumennya. Alur pembuatan dan verifikasi tanda tangan digital pada umumnya dapat dilihat pada Gambar 2.1.



Gambar 2.1. Alur Tanda Tangan Digital

C. Elliptic Curve Digital Signature Algorithm

Elliptic Curve Cryptography (ECC) adalah sebuah kriptografi kunci publik. ECC pertama kali dikembangkan oleh Neal Koblitz dan Victor S. Miller pada tahun 1985. ECC dibuat dengan tujuan memperpendek kunci tanpa mengurangi keamanannya. Dengan kunci yang lebih pendek, maka dibutuhkan memori dna komputasi yang lebih sedikit. ECC didasarkan dari perhitungan struktur aljabar dari kurva eliptik pada medan terbatas. Pada ECC, dibuat sebuah persamaan kurva seperti pada Persamaan (1).

$$y_2 = x_3 + ax + b \quad (1)$$

Penerapan ECC dalam tanda tangan digital mengembangkan sebuah pendekatan digital signature dengan nama Elliptic Curve Digital Signature Algorithm (ECDSA). Pada algoritma ini, dibutuhkan sebuah perjanjian terlebih dahulu di antara kedua belah pihak. Mereka perlu mengetahui parameter kurva (*CURVE*, *G*, *n*) di mana *CURVE* merupakan kurva yang digunakan, *G* adalah titik dasar orde bilangan prima pada kurva, dan *n* adalah orde perkalian titik *G*.

Untuk membuat kunci publik, kunci privat d_A akan dibuat dari sebuah angka acak. Kemudian, titik G akan dikalikan dengan d_A untuk menghasilkan kunci publik Q_A dalam bentuk titik.

Pembuatan tanda tangan digital menggunakan ECDSA dimulai dari mengubah pesan menjadi bentuk *hash E* dengan fungsi *hash* tertentu. Setelah pesan diubah menjadi bentuk *hash*, dipilih sebuah angka acak *k* dengan rentang 1 hingga n-1. *k* akan dikalikan dengan titik *G* untuk menghasilkan sebuah titik baru (x_1, x_2). Setelahnya, dicari nilai *r* dengan Persamaan (2).

$$r = x_1 \bmod n \quad (2)$$

Jika *r* memiliki nilai 0, maka titik *k* akan kembali dipilih dan dilakukan perhitungan ulang. Jika titik *r* tidak 0, maka akan dihitung *s* dengan Persamaan (3).

$$S = k^{-1}(E + d_A * r) \bmod n \quad (3)$$

k^{-1} pada Persamaan (3) menandakan modular invers dari *k*. Hasil tanda tangan digital adalah pasangan (*r*, *s*).

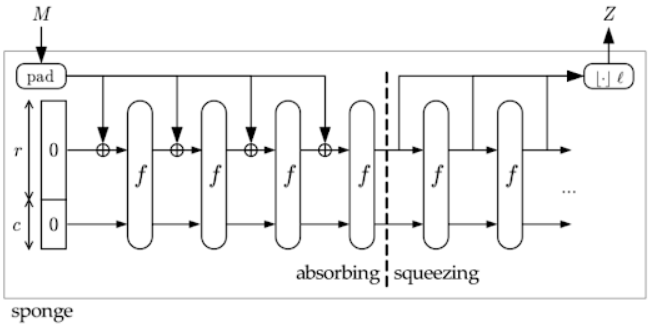
Untuk melakukan verifikasi tanda tangan digital, akan dilakukan tahap-tahap sebagai berikut. Pesan yang akan diverifikasi akan kembali diubah menjadi bentuk *hash E*. Setelah itu, akan dihitung sebuah titik (P_x, P_y) dengan Persamaan (4)

$$P_x, P_y = (s^{-1} * E * G) + (s^{-1} * r * Q_A) \quad (4)$$

Jika P_x memiliki nilai yang sama dengan *r*, maka tanda tangan digital pada pesan tersebut memiliki status valid.

C. SHA-3

Pada tahun 2007, National Institute of Standards and Technology (NIST) menyelenggarakan kompetisi terbuka untuk mengembangkan fungsi *hash* baru yang bernama SHA-3. Fungsi *hash* ini menjadi komplementer SHA-1 dan SHA-2. Pada tahun 2012, kompetisi berakhir dengan pemenangnya adalah algoritma Keccak. Berbeda dengan finalis SHA-3 lainnya, Keccak menggunakan konstruksi spons dengan fungsi non-kompresi untuk menyerap dan memeras *digest*. Seperti terlihat pada Gambar 2.3., terdapat tiga tahap pada algoritma Keccak, yaitu *preprocess*, penyerapan (*absorbing*), dan pemerasan (*squeezing*).



Gambar 2.2. Tahap Algoritma Keccak

Pada algoritma Keccak, terdapat *b*-bit peubah status (*state*) S

yang terdiri dari *r*-bit pertama yang akan dikenai operasi XOR dan *c*-bit terakhir yang tidak dikenai operasi XOR. Pada tahap *preprocess*, pesan *M* ditambahkan terlebih dahulu dengan bit-bit pengganjal (*padding*) menjadi *string P* sehingga setelahnya *P* dapat dibagi menjadi blok-blok yang setiap bloknya memiliki panjang *r*-bit. Setelah *P* dibagi menjadi blok-blok P_i, dilakukan inisiasi *state S* dengan nilai 0.

Pada tahap penyerapan, *r*-bit pertama dari *state S* dikenai operasi XOR dengan blok P_i yang juga memiliki panjang *r*-bit. Setelah itu, seluruh *state S* dimasukkan ke dalam fungsi permutasi *f*. Fungsi permutasi *f* adalah fungsi permutasi yang menggunakan operasi XOR, AND, dan NOT. Fungsi ini didefinisikan untuk setiap *power-of-two word size*, $w = 2\ell$ bits. Jika menggunakan 64-bit *words*, maka ℓ bernilai 6. Pada dasarnya, fungsi permutasi *f* terdiri atas $12 + 2\ell$ pengulangan dari lima tahap seperti pada *pseudocode* berikut:

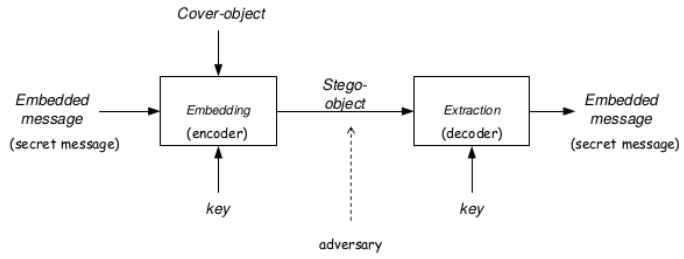
```
# θ step
C[x] = A[x,0] xor A[x,1] xor A[x,2] xor A[x,3] xor A[x,4],   for x in 0…4
D[x] = C[x-1] xor rot(C[x+1],1),                             for x in 0…4
A[x,y] = A[x,y] xor D[x],                                    for (x,y) in
                                                                 (0…4,0…4)

# ρ and π steps
B[y,2*x+3*y] = rot(A[x,y], r[x,y]),                          for (x,y) in
                                                                 (0…4,0…4)

# χ step
A[x,y] = B[x,y] xor ((not B[x+1,y]) and B[x+2,y]),           for (x,y) in
                                                                 (0…4,0…4)

# ι step
A[0,0] = A[0,0] xor RC

return A
```

Setelah tahap penyerapan selesai, dilakukan tahap pemerasan. Pertama-tama, pada variable *Z* yang akan menyimpan *message digest* hasil *hash*, akan dilakukan inisialisasi dengan *string* kosong. Selagi panjang *Z* belum sama dengan panjang *output* yang diinginkan, *r*-bit pertama dari *state S* disambungkan ke *Z*. Jika panjang *Z* masih belum mencapai panjang *output* yang diinginkan, *state S* dimasukkan terlebih dahulu ke dalam fungsi permutasi *f* sebelum proses diulangi kembali.

D. Steganografi

Steganografi adalah suatu teknik untuk menyembunyikan keberadaan (*existence*) dari sebuah pesan. Hal ini dilakukan untuk menghindari kecurigaan dari pihak ketiga. Steganografi termasuk dalam bidang ilmu *Information Hiding* bersama kriptografi. Berdasarkan ranah operasinya, steganografi dapat dibagi menjadi dua kelas:

- *Spatial domain methods*
- *Transform domain methods*

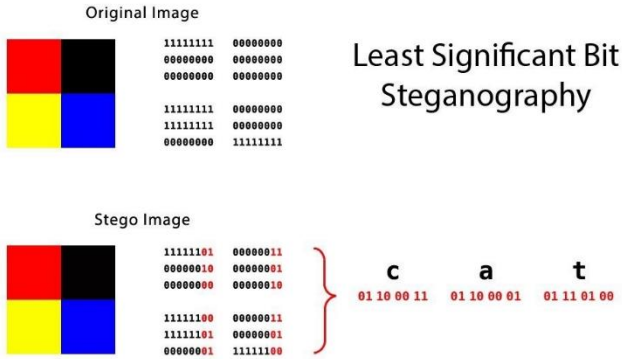


Gambar 2.3. Proses steganografi

Alur melakukan steganografi dapat dilihat pada Gambar 2.3. Tahapan yang dilakukan pada steganografi adalah memasukkan sebuah *secret message* yang dapat berupa file maupun teks ke dalam *cover-object*. Dapat dilakukan enkripsi terlebih dahulu pada pesan untuk meningkatkan keamanan. Hasil yang didapat adalah sebuah *stego-object*, yaitu *cover-object* yang sudah disisipi dengan pesan tertentu. Steganografi dapat dilakukan pada citra digital, audio, video dan media lainnya.

E. Metode LSB

LSB adalah teknik yang umum digunakan dalam enkripsi dan dekripsi informasi rahasia. Cara kerja metode LSB yaitu mengubah bit redundan dari *cover image* yang tidak dengan signifikan mempengaruhi bit dari pesan rahasia. Perubahan bit LSB ini tidak akan memberikan pengaruh terhadap persepsi visual / memori.



Gambar 2.4. LSB

Setiap pesan dapat disisipkan pada bagian LSB dari sebuah *cover-object* secara sekuensial atau secara acak. Besar pesan yang dapat disisipkan tergantung pada besar *cover-object*. Misalkan pada citra *grayscale* (1 *byte*/*pixel*) 256 x 256 *pixel*:

- Jumlah *pixel* = jumlah *byte* = 256 x 256 = 65536
- Jika setiap *byte* dapat menyembunyikan 1 bit pesan di LSB-nya, maka ukuran maksimal pesan = 65536 bit = 8192 *byte* = 8 KB

Ukuran pesan yang disembunyikan dapat ditingkatkan dengan menyembunyikan lebih dari 1 bit pada setiap *byte cover-object*. Namun, hal tersebut dapat menurunkan kualitas dari *stego-object* yang dihasilkan.

III. ANALISIS DAN RANCANGAN

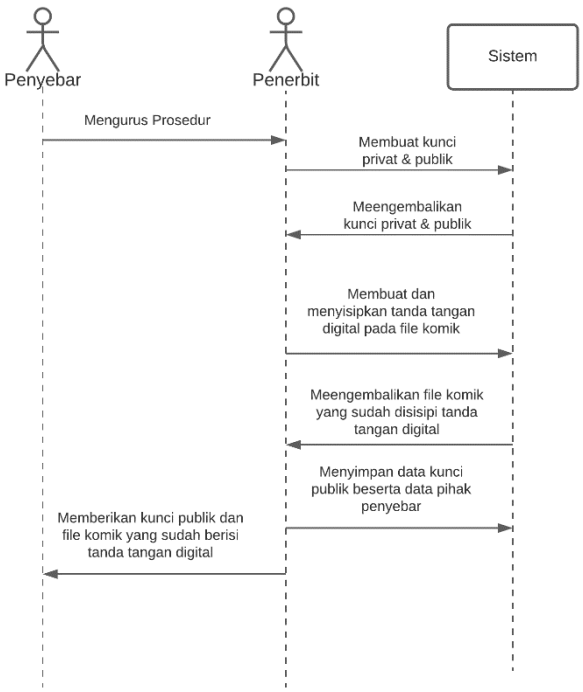
Pada bagian ini, dirancang sebuah sistem untuk melakukan verifikasi status legal komik dengan memanfaatkan tanda tangan digital. Secara umum, alur yang dirancang dapat dilihat pada Gambar 3.1. dan Gambar 3.2. Pada awalnya, pihak yang ingin melakukan penyebaran komik (untuk selanjutnya akan disebut sebagai pihak penyebar) melakukan permintaan kepada penerbit. Penerbit dan pihak penyebar lalu melakukan prosedur-prosedur pemberian izin. Setelah pihak penyebar mendapatkan izin, penerbit lalu melakukan pembuatan dan penyisipan tanda tangan digital dengan kunci privat khusus untuk pihak tersebut pada seluruh file gambar komik yang akan diberikan. Setelah tanda tangan digital selesai disisipkan pada seluruh file komik

yang akan diberikan, file komik tersebut beserta kunci publik diberikan kepada pihak penyebar.

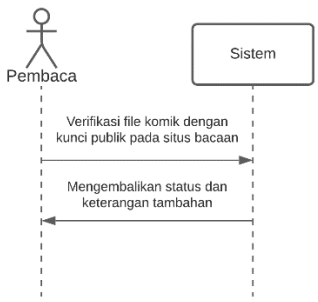
Pihak penyebar yang sudah mendapatkan file gambar komik dan kunci publik dapat mengunggah file gambar tersebut pada platform yang ia miliki. Kunci publik harus turut diunggah agar nantinya pembaca dapat melakukan verikiasi terhadap komik digital tersebut.

Agar verifikasi dapat dilakukan, dibutuhkan sebuah situs yang dapat digunakan oleh pihak yang ingin melakukan verifikasi. Situs ini akan menerima file gambar komik beserta kunci publik yang didapatkan dari platform penyebar komik. Situs tersebut lalu akan memberikan hasil apakah gambar terverifikasi legal atau tidak.

Dengan menambahkan fungsi pendataan kunci publik beserta penerbit dan pihak penyebar yang diberikan kunci publik tersebut, situs juga dapat memberikan hasil pada situs/aplikasi apa gambar tersebut seharusnya berasal. Hal ini dapat digunakan untuk mencegah pencurian komik dari situs legal yang kemudian disebarkan pada situs ilegal.



Gambar 3.1. Alur Rancangan Penyebaran Komik Digital



Gambar 3.2. Alur Rancangan Verifikasi Komik Digital

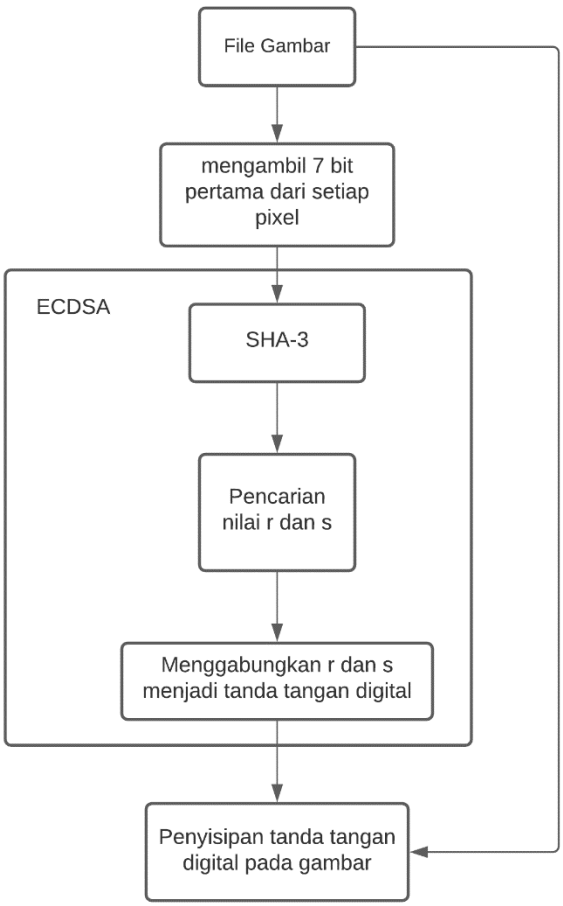
Untuk dapat melakukan hal-hal yang disebutkan sebelumnya, diperlukan 3 fungsi utama pada sistem yang akan dibuat, yaitu pembuatan kunci publik dan privati, pembuatan dan penyisipan tanda tangan digital, dan verifikasi gambar.

A. Pembuatan Kunci Publik dan Privat

Pada makalah ini, pembuatan kunci publik dan privat akan menggunakan Elliptic Curve Cryptography. Pada sistem, akan ditetapkan terlebih dahulu nilai tetap dari konstanta *G* sebagai titik dasar orde bilangan prima pada kurva dan *n* sebagai orde perkalian titik *G*. Setelah itu, kunci privat d_A akan didapat dengan membuat sebuah bilangan bulat secara acak. Kemudian, dilakukan perkalian antara titik *G* dengan kunci privat d_A untuk menghasilkan kunci publik Q_A yang berupa titik. Nantinya, kunci privat d_A akan disimpan oleh penerbit dan kunci publik Q_A akan diberikan kepada pihak penyebar untuk kemudian diunggah pada *platform* yang ia miliki.

B. Pembuatan dan Penyisipan Tanda Tangan Digital

Alur pembuatan dan penyisipan tanda tangan digital dapat dilihat pada Gambar 3.3.



Gambar 3.3. Alur Pembuatan dan Penyisipan Tanda Tangan Digital

Mula-mula, sistem akan menerima file gambar. Setelah itu, dilakukan *preprocess* untuk mengambil 7 bit pertama dari setiap *pixel* pada gambar tersebut. Bit-bit tersebut lalu digabungkan dan akan memasuki proses pembuatan tanda tangan digital.

Pada makalah ini, pembuatan tanda tangan digital akan menggunakan ECDSA (Elliptic Curve Digital Signature

Algorithm). Pada ECDSA, dilakukan pengubahan bit-bit ke dalam bentuk *hash* terlebih dahulu. Fungsi *hash* yang digunakan adalah SHA-3 256 bit (Keccak). Bit-bit yang didapatkan dari *preprocess* ditambahkan bit-bit pengganjal (*padding*) terlebih dahulu. Kumpulan bit ini disebut sebagai *P*. Selanjutnya, diinisiasi *state* sepanjang *r* + *c* bit (nilai *r* dan *c* telah ditentukan di awal). *P* kemudian dibagi menjadi blok-blok yang setiap bloknya memiliki panjang *r*-bit. *P* yang telah dibagi menjadi blok-blok P_i dimasukkan ke dalam tahap penyerapan. Pada tahap ini, *r*-bit pertama dari *state* dikenai operasi XOR dengan blok P_i yang memiliki panjang yang sama. Setelah itu, *state* dimasukkan ke dalam fungsi permutasi *f*. Proses tersebut diulang hingga seluruh blok P_i telah digunakan. Setelah selesai dari tahap penyerapan, dilakukan tahap pemerasan. Pada tahap ini, akan diambil *r*-bit pertama dari *state*. Jika sudah mencapai 256 bit, maka tahapan dihentikan. Jika belum, *state* akan masuk ke fungsi permutasi *f* dan diambil kembali *r*-bit pertama dari *state* untuk ditambahkan dengan bit yang telah diambil sebelumnya. Hasil dari fungsi *hash* ini untuk selanjutnya disebut sebagai *E*.

Proses selanjutnya memakai kunci publik dan kunci privat yang didapatkan sebelumnya. Kunci publik Q_A merupakan sebuah titik yang terdiri dari (x_1, x_2). Proses dimulai dengan mencari nilai *r* dari Persamaan (5).

$$r = x_1 \bmod n \tag{5}$$

Setelah didapatkan nilai *r*, dicari nilai *s*, dengan Persamaan (6).

$$s = k^{-1}(E + d_A * r) \bmod n \tag{6}$$

Hasil dari tanda tangan digital adalah pasangan (*r*,*s*) yang memiliki panjang 128 karakter atau 1024 bit.

Tanda tangan digital yang sudah dihasilkan akan disisipikan ke file gambar komik dengan cara steganografi. Metode steganografi yang dipakai pada makalah ini adalah metode LSB. Tanda tangan digital yang memiliki panjang 1024 bit ditambahkan bit-bit pengganjal terlebih dahulu hingga panjangnya mencapai banyak *pixel* pada file gambar. Setelah ditambahkan bit-bit pengganjal, barulah tanda tangan digital disisipkan ke dalam gambar. Karena memakai metode LSB, seluruh bit terakhir dari setiap *pixel* diganti dengan tanda tangan digital. Dengan begitu, tanda tangan digital dapat disisipkan tanpa mengubah tampak file gambar secara signifikan.

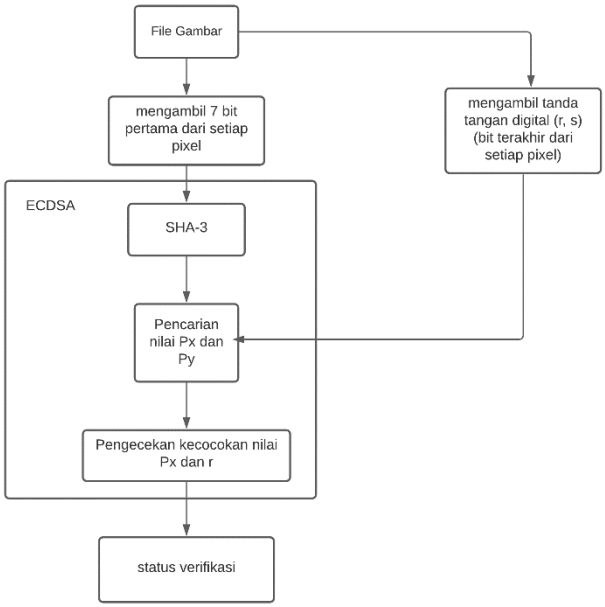
C. Verifikasi Gambar

Alur umum verifikasi gambar dapat dilihat pada Gambar 3.4. Untuk melakukan verifikasi gambar, pertama-tama dipisahkan terlebih dahulu tanda tangan digital dan bit-bit asal. Tanda tangan digital diekstraksi dari file gambar dengan cara mengambil seluruh bit terakhir dari setiap *pixel* dan melakukan *preprocess* untuk menghilangkan bit-bit pengganjal. Seperti pada saat pembuatan tanda tangan digital, bit-bit asal didapatkan dengan mengambil 7 bit pertama dari setiap *pixel* pada file gambar.

Tanda tangan digital merupakan pasangan (*r*,*s*). Verifikasi tanda tangan digital tersebut dilaukan dengan mengubah bit asal menjadi bentuk *hash E* seperti pada saat pembuatan tanda tangan digital. Selanjutnya, dilakukan pencarian sebuah titik P_x, P_y dengan Persamaan (7).

$$P_x, P_y = (s^{-1} * E * G) + (s^{-1} * r * Q_A) \tag{7}$$

Jika P_x memiliki nilai yang sama dengan *r*, maka tanda tangan digital telah terverifikasi valid. Setelah itu, sistem akan mencari dari *database* kepemilikan dari kunci publik yang menjadi masukan. Sistem lalu mengeluarkan pesan bahwa gambar telah terverifikasi status legalnya dan memberitahu pada situs / aplikasi apa gambar tersebut seharusnya berada. Dengan begitu, apabila gambar telah terverifikasi status legalnya namun situs / aplikasi berbeda dari seharusnya, pengguna dapat mengetahui bahwa gambar tersebut merupakan hasil pencurian dari tempat seharusnya.



Gambar 3.3. Alur Verifikasi Gambar

IV. KESIMPULAN

Dari sistem yang telah dirancang, didapatkan kesimpulan sebagai berikut :

1. Sistem verifikasi status digital dapat dibuat dengan memanfaatkan teori tanda tangan digital dan steganografi.
2. Dengan sistem yang dirancang, pembaca dapat mengetahui status legal dari komik yang ada pada situs / aplikasi yang digunakan untuk membaca komik digital.
3. Tidak hanya pembaca, penerbit juga dapat melakukan *tracking* jika terdapat pencurian dari komik yang menjadi haknya.

V. SARAN

Pada saat ini, masih sulit ditemukan sistem tanda tangan digital yang *robust* terhadap kompresi terutama jika gambar didapatkan dengan cara *screenshot*, sedangkan akan lebih

mudah bagi pembaca apabila verifikasi dapat dilakukan dari hasil *screenshot*. Karena itu, jika sudah terdapat sistem tanda tangan digital yang *robust* terhadap *screenshot*, maka sistem yang sudah dirancang pada penelitian ini dapat diperbarui agar pembaca dapat lebih mudah melakukan verifikasi.

VII. Acknowledgment

Penulis ingin berterima kasih kepada Tuhan yang Maha Esa atas berkat dan anugerah-Nya sehingga penulis dapat memiliki kesempatan untuk membuat sebuah makalah yang berkaitan dengan penerapan kriptografi pada kehidupan sehari-hari. Penulis juga ingin berterima kasih kepada Institut Teknologi Bandung yang telah memfasilitasi kegiatan belajar mengajar kami. Penulis berterima kasih kepada dosen khususnya kepada bapak Rinaldi Munir selaku dosen pengajar IF 4020 Kriptografi yang telah mengajari penulis mengenai Kriptografi. Dan yang terakhir, penulis berterima kasih kepada keluarga dan teman-teman kami yang mendukung kami dalam pembuatan makalah ini.

Referensi

[1] G. Bertoni, dll, *Keccak Specifications Summary*, Belanda: Radbound University, 2020. Diakses pada: 20 Desember 2020. [Online]. Tersedia: https://keccak.team/keccak_specs_summary.html

[2] R. Munir, *Bahan Kuliah IF4020 Kriptografi SHA-3 (Keccak)*, Bandung: Institut Teknologi Bandung, 2020. Diakses pada: 20 Desember 2020. [Online]. Tersedia: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/SHA-3-2020

[3] R. Munir, *Bahan Kuliah IF4020 Kriptografi Steganografi (Bagian 1)*, Bandung: Institut Teknologi Bandung, 2020. Diakses pada: 20 Desember 2020. [Online]. Tersedia: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Steganografi-Bagian1-2020.pdf

[4] R. Munir, *Bahan Kuliah IF4020 Kriptografi Steganografi (Bagian 2)*, Bandung: Institut Teknologi Bandung, 2020. Diakses pada: 20 Desember 2020. [Online]. Tersedia: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Steganografi-Bagian2-2020.pdf

[5] R. Munir, *Bahan Kuliah IF4020 Kriptografi Tanda-tangan Digital*, Bandung: Institut Teknologi Bandung, 2020. Diakses pada: 20 Desember 2020. [Online]. Tersedia: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Tanda-tangan-digital-2020.pdf

[6] Shueisha, *Manga Plus by Shueisha*, Jepang, 2019. Diakses pada: 20 Desember 3030. [Online]. Tersedia: https://mangaplus.shueisha.co.jp/

[7] Sudjana, dll, *Media Pengajaran*. Bandung: Sinar Baru Algensino, 2015.

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, 21 Desember 2020

Fithratulhay Pribadi
13517140